PROCEEDINGS
OF THE
SIXTH SEMINAR
ON THE
DOD COMPUTER SECURITY
INITIATIVE

NATIONAL BUREAU OF STANDARDS
GAITHERSBURG, MARYLAND

NOVEMBER 15-17, 1983

The following memo from the Under Secretary of Defense, Richard G. Stillwell, was presented as an introduction to the DoD Computer Security Center and the DoD Trusted Computer System Evaluation Criteria. It authorizes and encourages use of the Criteria in evaluating the needs of the various DoD components.

The DoD Trusted Computer System Evaluation Criteria is available upon request from the Center. The address is:

DoD Computer Security Center

9800 Savage Rd.

Ft. George G. Meade, MD 20755

THE UNDER SECRETARY OF DEFENSE

WASHINGTON, D.C. 20301

POLICY

15 NOV 1983

In reply refer to
I-12027/83

MEMORANDUM FOR SECRETARIES OF THE MILITARY DEPARTMENTS
CHAIRMAN OF THE JOINT CHIEFS OF STAFF
DIRECTORS OF THE DEFENSE AGENCIES
DIRECTOR, WASHINGTON HEADQUARTERS SERVICES

SUBJECT: Security Requirements for ADP Systems -
DoD Trusted Computer System Evaluation Criteria

References: (a) DoD Directive 5200.28, "Security Requirements for Automatic Data Processing (ADP) Systems," December 18, 1972, as amended

(b) DoD Manual 5200.28-M, "ADP Security Manual," January 1973, as amended

(c) DoD Directive 5215.1, "Computer Security Evaluation Center," October 25, 1982

References (a) and (b) promulgate policy and assign responsibilities for the security analysis, test, evaluation and approval of ADP systems proposed for the processing of classified information. As most of you are aware, the primary impediment to the cost effective implementation of that policy, particularly the overall system security evaluation/approval process, has been the lack of technical hardware/software security criteria and evaluation methodologies.

In response to this long standing need and to build upon previous DoD developmental efforts that focused on secure ADP systems technology, the DoD Computer Security Evaluation Center was established in January 1981. The Center's charter, DoD Directive 5215.1 (reference (c)), specifically tasks the organization to "...complement the established responsibilities of DoD Components relating to overall policy, security evaluation, and approval of computer systems as prescribed in DoD Directive 5200.28, DoD 5200.28-M..." and others, by establishing and maintaining "...technical standards and criteria for the evaluation of trusted computer systems that can be incorporated readily into the DoD Component life-cycle management process..."

The enclosed document, "Department of Defense Trusted Computer System Evaluation Criteria," August 15, 1983, has been issued in direct response to that tasking. Accordingly, pursuant to responsibilities assigned to me for overall security policy, standards and criteria applicable to ADP systems processing classified information, I hereby authorize and encourage use of the attached evaluation criteria document in meeting your responsibilities assigned by references (a) and (b) on an interim basis, pending full formal coordination of the document.

The Trusted Computer System Evaluation Criteria should serve as the prime reference document whenever automated systems technical protection issues are addressed, and it should be used with specific regard to the following:

1. The criteria provide an excellent means for measuring the current technical security posture of your systems against potential enhanced security capabilities. This document should accordingly be used in the performance of security evaluation activities conducted to assess current computer system security effectiveness pursuant to responsibilities assigned by reference (a);

2. The document provides detailed descriptions of criteria for meeting basic technical computer security requirements associated with the processing of classified and other sensitive information. Accordingly, the criteria should be used during the design phase of the system life-cycle for the formulation and specification of security requirements for systems under development and for future systems; and,

3. The criteria set forth in the attachment are essentially independent of any specific vendor's product line. Accordingly, they should be used in procurement specifications to establish a minimum acceptable level of hardware/software protection features.

Richard G. Stilwell
General, USA (Ret.)
Deputy

1 Attachment a/s

cc: Under Secretaries of Defense
Assistant Secretaries of Defense
Assistants to the Secretary of Defense

CONTENTS

TECHNICAL PROGRAM

Conference Theme: Trends in Computer Security

Tuesday, November 15

Trend 1 - Identifying the Spectrum of Computer Security Requirements

Welcoming Address
James Burrows
Director
Institute for Computer Science and Technology
National Bureau of Standards

Conference Introduction
Melville Klein
Director
DoD Computer Security Center

Keynote Speaker
Hon Richard D. DeLauer
Under Secretary of Defense for Research and Engineering

Government Policy and Legislative Initiatives Bearing on Computer Security
Louise Becker
Library of Congress
Congressional Research Service

Computer Security and the Federal Central Management Agencies
Edward Springer
Office of Management and Budget

A Legislative Perspective on Computer Security
Anthony C. Taylor
Staff Director
Committee on Science and Technology, Subcommittee on Transportation, Aviation and Materials

An Industry View of the DoD Computer Security Center Program
Robert H. Courtney, Jr.
President
Robert H. Courtney, Inc.

Panel Session -- Base Spectrum of Computer Security Requirements
Dr. Stuart Katzke, Moderator

Panel Members
James P. Anderson - James P. Anderson & Co. **(TEXT NOT AVAILABLE)**
William H. Murray - IBM Inc.
Nancy Woolsey - Lockheed
Jimmie E. Haines - Boeing Computer Service Co.

Trusted Computer System Evaluation Criteria: Major Divisions
Sheila Brand
Chief, Standards
DoD Computer Security Center

Trusted Computer System Evaluation Criteria Classes B1 through B3
Daniel J. Edwards
Chief, Standards and Products
DoD Computer Security Center

Requirements for Class A1 Systems and Major Differences Between Division A and Division B Systems
Dr. Carl Landwehr
Naval Research Laboratory

Panel Session -- Security Requirements for Computer Networks and Professional Workstations
David E. Bell, Moderator

Panel Members
Ray McFarland - DoD Computer Security Center
John White - The MITRE Corp.

Wednesday, November 16

Trend 2 - Transferring Computer Security Technology from Laboratory to Marketplace

Keynote Speaker
Barry Schrager
President
SKK,Inc.

Beyond A1: The Research and Development Challenge
George Jelen
Chief, Research and Development
DoD Computer Security Center

Computer Security Research and Development
Howard Weiss
DoD Computer Security Center

Computer Network Security
Dr. Dennis Branstad
Institute for Computer Science and Technology
National Bureau of Standards **(TEXT NOT AVAILABLE)**

DoD Computer Network Security: Projects and Projections
Col. John Lane
Director
Information Systems Division, C^3I

Computer Network Security: Public and Private
Dr. Stephen Kent
Computer Security Specialist
BBN,Inc.

Security Mechanisms in the Livermore Interactive Network Communication System (LINCS)
Dan Nessett
Lawrence Livermore National Laboratories

Computer Network Security Research
Paul Cook
Ford Aerospace Inc. **(TEXT NOT AVAILABLE)**

Verification Technology Transfer
Richard Kemmerer
Professor
University of California, Santa Barbara

The Formal Development Methodology (FDM)
Deborah Cooper
System Development Corporation

The Hierarchical Development Methodology
Peter Neumann
SRI International

The Gypsy Verification Environment
Donald Good
University of Texas **(TEXT NOT AVAILABLE)**

Panel Session -- Security Technology in the Marketplace
Dennis Steinauer, Moderator

Panel Members
Peter S. Browne - EDP Audit Controls, Inc.
F. Lynn McNulty - U.S. Department of State

Randy N. Sanovic - Mobil Corporation
Eddie L. Zeitler - Security Pacific Bank

Thursday, November 17

Trend 3 - Emergence and Evaluation of Specific Computer Security Products

Keynote Speaker
Stephen Walker
President
Trusted Information Systems, Inc.

Factors in Evaluating Computer Security
Zella Ruthberg
National Bureau of Standards

Evaluation, the DoD Certification Process, and its Relation to the Trusted Computer System Security Criteria
William Neugent
The MITRE Corporation

Security Requirements, Control Objectives, and the Role of the Auditor
Courtland Reeves
Peat, Marwick, Mitchell & Co.

Civilian Agency Views of Evaluation, Certification and Accreditation, Part 1
Lillian Duffey
Federal Emergency Management Agency

Civilian Agency Views of Evaluation, Certification and Accreditation, Part 2
Fred Tompkins
NASA

Panel Session -- Available Computer Security Products Satisfying Stated Requirements
Mario Tinto, Moderator

Panel Members
Linda Vetter - SKK,Inc.
Stan Kurzban - IBM,Inc.
Terry Cureton - Control Data Corporation
Benson Margulies - Honeywell
Paul Cudney - System Development Corporation
Lester Fraim - Honeywell

Evaluation Process and Problems
Paul Woodie
Chief, Commercial Products Evaluations Team
DoD Computer Security Center

Computer System Security Testing
Maj. Douglas Hardie
DoD Computer Security Center

Evaluations of Applications Systems
Suzanne O'Connor
DoD Computer Security Center

Panel Session -- How Do You Sell Better Computer Security?
Steven Lipner, Moderator

Panel Members
Lester Fraim - Honeywell
Theodore Lee - UNIVAC
Steven Lipner - Digital Equipment Corporation

WELCOMING ADDRESS

James Burrows

Director

Institute for Computer Science and Technology

National Bureau of Standards

We have been planning this Conference for over a year, and it is coincidental that it is being held at a time when computer security problems are receiving widespread attention in the press and in the government.

Computer security has been discussed for nearly 2Ø years, especially in the defense and intelligence communities. The Institute for Computer Sciences and Technology established its computer security program in 1973 to help the non-defense community and private sector computer users meet their responsibilities for computer and data security. Computer security had a high priority at that time and it remains a vital component of our program. It is very clear, however, from a recent press report on break-ins to computer systems that computer security is still a problem - a problem which has not changed much over the past 2Ø years. It is a problem that demands our attention as managers, but it is not cause for panic. Many of the solutions to the problem are available, but we need to stimulate a higher level of awareness of both the problems and solutions.

In recent weeks, the Congress and other government officials have heard just how vulnerable many computer systems are to intentional destructive acts and how easy it is to penetrate them; however, the computer security problem includes the threats, vulnerabilities, and potential losses that can occur from within our own personnel areas as well as from intentional acts from outside. The hearings produced some interesting recommendations for augmenting our role in computer security. It was suggested that we operate a federal center offering direct support to all non-defense agencies, which is being done; that we initiate the development of security standards for all computer terminals manufactured in the United States and abroad, which is something that is not currently being done; and that we develop a manual of technical security features that would be mandatory for federal computer acquisitions, which we try to do.

Between 8Ø and 9Ø percent of current computer security problems can be addressed by well-defined, cost-effective, and available solutions, including management and technical procedures. We've worked at these in the past and we will continue to work with many organizations that are active in developing these policies and implementing solutions for computer security. The Defense Department, OMB, Congressional Staff, Congressional Research Service, NASA, and Department of Justice are some of these organizations. Through the cooperation of these organizations, we identify problems common to all and seek solutions that will satisfy the fundamental requirements of all these communities.

Drawing on that work, we have developed a list of activities that form the basis for a comprehensive security program. This is one approach to achieving a high level of awareness. On one of the handouts you could have gotten at the door, we have broken these security activities into four major groups: The first group is Policy and Administrative Activities - these should be made mandatory at all organizations. They include establishing an organizational security policy, selecting employees carefully, ensuring awareness of computer security among users, managers, operators, system and security personnel, providing appropriate employee training and performing risk analyses. Based on those risk analyses, activities in the other three groups of activities should be selected to achieve a balanced program. These structured activities of all four groups are listed in the handout.

While we don't specifically mention the security requirements of small computer systems, we believe that the activities listed apply to the small systems as well. As small systems are linked together into networks, their computer security requirements will be similar to those of large- or medium-scale systems.

Technical solutions beyond what we have recommended will become increasingly important with the increasing use of computer networks, distributed data processing, satellite transmission of data, and national/international electronic fund transfers. Trusted System Architectures, which you will be discussing in this meeting, and other technical safeguards will be the key to managing and controlling our data processing activities in the future. By working together, I believe we can develop more effective solutions to one of our most serious information processing problems. In the meantime, it is essential that we use the solutions that we have.

We are pleased to co-sponsor this conference with the DoD Computer Security Center, which has made excellant progress in addressing technical computer security problems.

CONFERENCE INTRODUCTION

Melville H. Klein

Director

DoD Computer Security Center

Good morning and welcome to the 6th Computer Security Conference. The Computer Security Center is pleased to join again with NBS in co-sponsoring this event.

A prime objective of this Conference is to show that computer security is only part of the proverbial weather problem. It is certainly one that everyone has been talking and writing about, but it is also one that those assembled here have been doing something very positive about.

Computers, as the single most ubiquitous component of defense, C^3 and intelligence systems as well as in the control of modern weapon delivery systems, have become critical links to the successful prosecution of the defense mission. As such, their integrity must be assured.

With computer security fast becoming a leading growth market in the government and the private sector these conferences foster the kind of collaboration needed between DoD, civil agencies, industry and academe to address common concerns.

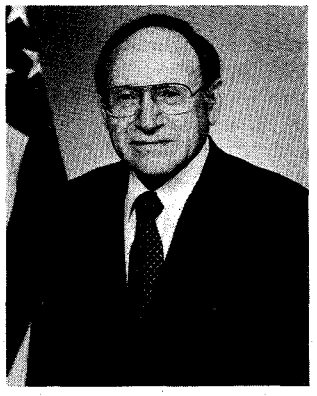
We are most fortunate this morning to have with us the Honorable Richard DeLauer, Under Secretary of Defense for Research and Engineering, our keynote speaker, to provide a DoD perspective on computer security. Dr. DeLauer, a Navy veteran, had a distinguished career with TRW before rejoining the Department of Defense in 1981. A Mechanical Engineering graduate of Stanford University, he took his B.S. degree from the Naval Postgraduate School, his A.E. and PHD degrees from Cal Tech, and has authored two books, *Nuclear Rocket Propulsion* and *Fundamentals of Nuclear Flight.*

Without further ado, the Honorable Richard DeLauer.

KEYNOTE SPEAKER Day 1
COMPUTER SECURITY REQUIREMENTS IN MAJOR DEFENSE PROGRAMS

Dr. Richard D. DeLauer

Under Secretary of Defense for Research and Engineering



Dr. Richard D. DeLauer was nominated by President Ronald Reagan to be Under Secretary of Defense for Research and Engineering on March 3, 1981. He was confirmed by the Senate on May 6, 1981 and sworn in on May 7, 1981.

As the USDRE, Dr. DeLauer is the principal advisor and assistant to the Secretary of Defense for Department of Defense scientific and technical matters; basic and applied research; development and acquisition of weapons systems; communications, command and control; atomic energy; and intelligence resources. He serves as the focal point for all test and evaluation matters. He is also the Defense Acquisition Executive (DAE).

Prior to his appointment as the Under Secretary, Dr. DeLauer was responsible for TRW Inc.'s Systems and Energy activities, which employed more than 20,000 people, and provided a wide variety of products and services for aerospace, electronic, industrial, civil and commercial markets.

He is a fellow of both the American Institute of Aeronautics and Astronautics and the American Astronautical Society. He is a member of the National Academy of Engineering, American

Association for the Advancement of Science, New York Academy of Science, Sigma Xi, the Engineering Advisory Council of the University of Southern California, the Advisory Committee of the Institute for the Advancement of Engineering, the Stanford Cabinet, and the Associates of the California Institute of Technology. He is Chairman of the LA Chamber of Commerce Aerospace Committee, founding Chairman of the Board of Governors of the American League for Exports and Security Assistance, and national chairman for Corporations of Stanford University.

Dr. DeLauer is the co-author of two books, Nuclear Rocket Propulsion, *and* Fundamentals of Nuclear Flight, *and has served as visiting lecturer at UCLA on nuclear rocketry.*

Dr. DeLauer graduated from Stanford University in 1940 with an A.B. in mechanical engineering. He received a B.S. in Aeronautical Engineering in 1949 from the U.S. Naval Postgraduate School, and an aeronautical engineering degree (A.E.) and a Ph.D. in Aeronautics and Mathematics from California Institute of Technology in 1950 and 1953 respectively.

Good morning, ladies and gentlemen. I'm especially happy to be here today at this sixth conference on computer security, because you are the folks from government and industry who have been working together to make better computer security happen.

Why do we need computer security? The answer to that question lies in why we need computers. I'm sometimes asked if the Department is becoming too heavily dependent on automation; if the DoD wouldn't be better off using less sophisticated technology to perform its mission. Well, clearly, the Department must use computers in many ways to accomplish its mission of national defense. First, we're a very large organization, composed of personnel in hundreds of units stationed at many locations around the globe. Thus, computers must be used by the Department to handle the routine personnel and logistics functions inherent in the operation of any organization of such size. Second, and more purely military, is the application of automation as an integral part of weapons systems themselves. For example, computers are used in missile guidance systems and in tank, artillery aircraft and ship fire control and other systems. Third, computers are essential parts of the decisionmaking process at theater and national levels. For example, we could not know of an impending missile attack upon the United States in sufficient time to take appropriate defensive measures without our automated electronic warning equipment. We could not engage in military operations against a technically sophisticated enemy without the use of automated electronic equipment. We can't do without computers, but they do present potential problems, against which we must guard. Computer security, or the protection of the information handled by these systems against unauthorized disclosure or modification, is one of those concerns.

We're not only dependent on computers, we're also dependent on the means for moving information to the people who need it. Effective data communications have become critical to the Department of Defense because information has become a resource as critical to the commander in the field as ammunition. Data communications networking is providing an answer to that challenge, but as with any new technology, it is bringing its own set of

problems, not the least of which is the network aspect of computer security. Networking has added a new dimension to the need for effective computer security, because it makes our computer systems more accessible to a would-be penetrator.

Not too long ago, we thought computer security was easy - if we thought about it very much at all. We believed that all we had to do was guard the computer room door, scan the output to make sure the computer hadn't mislabeled or interlaced the data, and be careful that we gave the data to the right persons when they came to the window to pick it up. I'm simplifying, of course - we also worried about things like emanations security. Generally, though, we applied traditional mehods of protecting data to our computer operations. And that wasn't a bad approach, in those days before data networking and massive reliance on processing power.

As computing matured, we did begin to recognize some shortfalls in the traditional approaches. The methods we employed, and continue to employ, impose restraints on computer use which restrict their full potential. A typical technique which we use, for example, is operating a computer in a "system high" mode, in which only personnel cleared to the highest level of classifed information processed by the computer can use the system or its products. Another similar technique frequently used is the "parallel processor" mechanism, in which one computer is used for unclassified work, while another is used for SECRET, another for TOP SECRET, and so on, with access to each system granted to only properly cleared personnel. "Periods Processing," in which the system is used for classified processing at one time during the day, and used for unclassified during another is often employed. The system is "purged" of all classified information prior to uncleared personnel's being granted access. All these techniques are effective, but they impose high dollar and operational cost. One of our major objectives is ease of information interchange, but the methods which we must employ today to secure our information effectively are operational impediments to the ease of information flow which our military requirements demand. Our ideal information system would allow totally secure simultaneous use of a processor for all levels of classification from UNCLASSIFIED through the most sensitive information and transmittal of that information through a network securely accessed by multiple users at different security levels. I mean, of course, a truly multi-level-secure processor operating into a truly multi-level-secure network, with ease of information interchange, between and among users at all security levels. We have a long way to go.

The length of that future journey has been more than underscored lately by the publicity given to hackers and the ease with which they seem to be able to penetrate our open unclassified networks. We're not surprised by that fact, of course. The Department is well aware of the security weaknesses of many commercial computer systems, and, through the DoD Computer Security Evaluation Center, is aggressively working with industry to make available more trusted systems. However, in a number of installations we currently have unclassified, remotely accessible systems with security controls that are not particularly strong. Thus, we are not really surprised if we discover that, as the result of illegal activities, there is unauthorized access to these systems. Within the limitations of these systems, we strive through good management and vigilance to minimize the vulnerabilities and risk. We avoid any possibility of significant damage to the national security by carefully limiting the information that is processed on these machines.

There is a major point to be made here. Although we're not surprised by the success of the hackers, the recent publicity has served to highlight our basic computer security dilemma - we know how to make our systems very, very secure, but to do so radically inhibits their use. Today, there is a trade-off between the cost of making an unclassified network more secure and the benefit of doing so. To make a network invulnerable to authorized users using today's security technology would be prohibitively expensive for unclassifed operations. To illustrate this point, a bank, if made invulnerable to an armed robbery, could likely not be conveniently used by the banking public. Thus, a bank employs reasonable security measures and society vigorously prosecutes the occasional armed robber. There is a cost to this way of doing business - because information is not classified does not mean that it has no value to a potential economic or military adversary, or to one seeking to illegally profit from unauthorized use of privileged information; we must develop good, solid, cost-effective computer security protections for all information, whether it is classified or unclassified.

We haven't just recognized this problem - you folks have been working it for years. The establishment of the DoD Computer Security Evaluation Center in 1981 as a pool of computer security excellence was a major expression of our community's awareness of the issues and of our intent to do something about the problems. That was a good start, but we need to do much more, and we're committed to doing it. We're mounting a major attack on the computer security problem on four fronts - policy, educational, administrative, and technical. I'd like to tell you a little bit about our intentions in each of these four areas.

First, Policy. Our basic computer security policy was developed during the computer era's equivalent to the Middle Ages - we must update our national policy to reflect and protect against the new vulnerabilities and risks imposed by the emerging network technologies. We are beginning to work this problem in conjunction with the other concerned government agencies. We must also encourage and support legislative initiatives to reduce our information vulnerablilities. As an example, current legislative proposals would clearly make unauthorized access to information a federal crime. Other legislative proposals are addressing various aspects of transborder data flow.

We also need to continue our policy of encouraging industry development of trusted computer systems. We in the Department of Defense are not alone in our need for secure systems. Other federal agencies and commercial enterprises have important privacy, security, and competitive considerations which cry for better data security. Independent Department of Defense development of secure systems would not, in the long run, provide affordable systems for all our requirements. All of us, government and private sector users, and computer manufacturers, must join in a partnership to advance the state of the art in

trusted systems. As part of our continuing dialog with industry in this regard, Secretary Weinberger, Deputy Secretary Thayer and myself will be meeting soon with the chief executive officers of many of the nation's leading computer and telecommunications companies.

As a matter of policy, we are exploring the establishment of a DoD-wide computer system security evaluation program. We need to do this to establish a clear baseline of our current security posture. Part of this effort, of course, will include the evaluation of the security aspects of commercially available products. The DoD Computer Security Evaluation Center is doing some selective evaluation work, but we need to systematically apply their efforts on a Department-wide basis. Development by the Center of Evaluation Criteria has been a major step forward, and we are in the process of formalizing and promulgating those criteria. We intend that new computer acquisitions and major upgrades of existing systems should clearly specify the degree of security required as part of the procurement specifications. The evaluation criteria now give us a tool to effectively specify the degree of required security.

We are also carefully looking at our information disclosure policy to ensure that we have a program which properly balances our commitment to give technical and other information the widest possible dissemination with our commitment to ensure that sensitive information remains in friendly hands. This is a difficult process in a free and open society, and yet we must exercise prudent judgment to balance these competing concerns as best we can.

Our second major front is educational - we must increase the awareness of all our people that information is a valuable resource, which must, as any valuable resource must, be carefully protected. To this end we are briefing senior decision makers on both the capabilities available and challenges inherent in protecting our automated information systems. For example, we have recently briefed both the Secretary and Deputy Secretary of Defense on the strengths and vulnerabilities of our computer systems, and on the protective measures we use to protect information. We also intend establishing a national level computer security course as an institutional means of ensuring that a continuing educational effort is carried on. We will also intensify our liaison with industry, so that our needs are more fully understood and become reflected in the commercial availability of features which improve computer security. Finally, we are incorporating computer security training as part of the standard curriculum of our military department computer training courses.

Our third campaign invokes administrative measures. We must see that our personnel use the security tools which are available to them today. We recognize that such measures as passwords have limitations, but when properly employed, they do form a first line of defense against unauthorized access. All too often, however, a user's password consists of his or her own last name, or some other easily determinable choice. We're not only failing to lock the door, we're leaving it wide open with an embroidered welcome mat. We must also establish more uniform procedures for audit trails and guidelines for procurement actions. We need to use our existing Computer Security Evaluation Center as a clearing house and dissemination point for vulnerability and other information. We're looking now at how to improve these essentially administrative processes.

Our fourth thrust is technical. The Computer Security Evaluation Center has the mission of consolidating and rationalizing the generic computer security program within the Department of Defense. We have given special attention to the research and development aspect as we have worked on the 1985 and beyond program, and I believe that when the budget cycle dust has settled, we will have a much stronger and accelerated computer security program. As part of that effort, we need to give increased attention to our existing means for encoding unclassified data, to make such systems as the data encryption standard more affordable. We clearly need to accelerate use within the department of commercially available systems which do exhibit strong computer security characteristics. I've heard it said that a DoD computer acquisition has never been won or lost on the basis of security features. We need to turn that perception around so that our friends in industry can know that we're really serious about security. We're also looking hard at our approach to acquisition of embedded computer systems, to ensure that any systems especially developed for military applications have security designed in from the start. That approach applies to all our R&D initiatives - we intend to ensure that computer security features are fully considered as DARPA helps us move into the next computing generation.

Well, I've tried to convey to you this morning that we in the Department of Defense are intent on improving the state of our computer security and that we're advancing on all fronts. In many respects, I've certainly been preaching to the choir - you folks are the pioneers who first began to recognize the problem and to do something about it. I suppose the best thing about preaching to the choir is being reasonably sure that your audience will agree with you. I'd like to test that hypothesis now by giving you folks the floor. Are there any questions you'd like to ask me?

I've certainly enjoyed the opportunity to come out here and to talk with you about the computer security concerns we share. You're doing a great job, but don't slow down yet. We still have a long way to go. Thank you, and I wish you a very successful conference.

GOVERNMENT POLICY AND LEGISLATIVE INITIATIVES BEARING ON COMPUTER SECURITY

Louise Becker

Library of Congress

This morning we begin a forward look at the complex subject of computer security, especially as it relates to Federal Government policies and current legislative initiatives. I would like to give you my perspective on how Congress has viewed this subject. From Capital Hill computer security has often been viewed from the following perspective: (1) improving/protecting National security and defense, this I believe brings most of you to this meeting, (2) developing effective computer/communications systems and improving Federal automated infromation resources, (3) limiting computer crime and abuse, and (4) protecting personal privacy and confidentiality of certain sensitive data.

Computer security, as you all know, is an evolving subject from both a technological and policy sense. I would like to provide you with some background, especially some of the critical events of the last twenty years.

In the late 196Øs, although computers had been with us two decades, there was little recognition of the computer security dimension. Attention was slow to focus on the fact that certain technological innovations which would permit us to directly access automated data and share computerized resources needed special protection. In that same decade, Congress enacted P.L. 89-3Ø6 (generally referred to as the Brooks Act). This landmark legislation focused attention on the effective management of automated data processing equipment and marks the attempt by Federal Government to begin management of this area. The Act calls for effective and efficient use of this technology and sets the framework on managing Federal computer use. Another important issue tackled by Congress in the 196Øs is the problem of personal privacy, especially as it might be eroded by the new technology.

In the 197Øs, Congress enacted legislation to protect against abuses in those systems handling personal data. In this time period we see the enactment of the Fair Credit Reporting Act, the Privacy Act of 1974, amendments to the Omnibus Crime Control and Safe Streets Act, and legislation creating the several national commissions to examine the problem of protecting information. Legislation creating the Privacy Protection Study Commission, Commission of Federal Paperwork, and the National Commission on Electronic Fund Transfers, called attention to the need to secure information. The Foreign Corrupt Practices Act, enacted in the 197Øs stimulated the private sector to consider appropriate attention to computer and communication security for certain information systems.

In the late 197Øs the Senate Committee on Government Operations, then chaired by Senator Abraham Ribicoff, began an examination of computer security and computer crime. As a result of the Committee's investigations into computer security Senator Ribicoff introduced the first "computer crime bill." Congressional concern and interest in protecting computerized systems has continued. Over the years Congress has encouraged better management of resources. Congress has urged that the Office of Management and Budget take the lead in improving the management of Federal computerized resources. Mr. Edward Springer, who follows me on this program will address the OMB's work in this area.

In the 197Øs the National Bureau of Standards issued the Data Encryption Standard (DES) which provided for an algorithm for encrypting non-national security sensitive data. This standard is used by both those agencies handling non-national security data which requires a certain level of protection and by the private sector.

Congress continues to be concerned with computer and communications systems security. In the 198Øs new laws, such as the Paperwork Reduction Act of 198Ø and the Financial Management Integrity Act, lay the foundation for improving the security of certain Federal Government systems. The establishment of the DoD Computer Security Center at the National Security Agency in 1982 provides a much needed focus for research and development in computer security.

In the 98th Congress legislative measures are pending and there have been a series of hearings both in the Senate and the House of Representatives on the subject. Among the bills pending are the following:

H.R. 1Ø92, *The Federal Computer Systems Protection Act of 1983*, introduced by Representative Bill Nelson of Florida. (This bill and others were the subject of hearings by the House Committee on the Judiciary, Subcommittee on Civil and Constitutional Rights.) An identical bill S. 1733 introduced in the Senate by Senator Trible has been referred to the Senate Committee on the Judiciary.

H.R. 3Ø75, *Small Business Computer Crime Prevention Act*, introduced by Representative Ron Wyden of Oregon. (Hearings were held by the Committee on Small Business in 1983 on this measure and subsequently it was passed by the House.) And an identical bill, S. 192Ø, *Small Business Computer Crime Prevention Act*, introduced by Senator Paul Tsongas of Massachusetts. Referred to the Senate Committee on Small Business.

H.R. 4384, a bill to establish a computer security research program and an Interagency Committee on Computer Crime and Abuse Task Force and provides criminal penalties for computer abuse, introduced by Representative Dan Mica of Florida. Referred to the House Committee on the Judiciary.

H.R. 43Ø1, a bill to amend Title 18 of the U.S. Code. Provides penalties for certain computer-related crime. Referred to House Committee on the Judiciary.

In addition to the measures pending before Congress, Committees of the House and Senate held hearings on the subject of computer security. These hearings grew out of concern for safeguarding computer resources, especially those in the national interest. In addition these hearings reviewed current events and assessed Government's role in securing information technology resources. The House Committee on Science and Technology Subcommittee on Transportation, Aviation and Materials held hearings on computer and communications security and privacy. Anthony Taylor, staff director of the Subcommittee, who is a member of this panel, will discuss the hearings in more depth.

The Senate Committee on Governmental Affairs Subcommittee on Oversight of Government Management held hearings in late October 1983. These hearings reviewed current actions and the recent reports of unlawful access to certain systems.

There are a number of problems which must be faced in computer security. Two important questions emerge which may require additional consideration: One is, what are the existing barriers to developing and implementing computer safeguards? The other is, what legislative measures are needed to protect computerized resources? Let me close by saying that we are at an important juncture in computer security. Congress needs to hear from you, the professionals, what the problems are and what solutions can help remedy the problems. I think it is important as professionals in computer security to establish a dialog on these critical issues concerning computer security. I think we can all agree that we live in a great society in which we can all participate in developing new directions and policies. Changes are being considered and options are being weighed at this time, therefore I urge you to comment on these matters to the Congress either as individuals or through your professional associations. We need an exchange of ideas on this complex subject so that we can be assured that we will reach the best solution for this nation.

COMPUTER SECURITY AND THE FEDERAL CENTRAL MANAGEMENT AGENCIES

Edward Springer

Office of Management and Budget

I'd like to talk briefly about some of the recent events we have all read about in the press and what the threats are that we see from these events. I would like to talk about how we in the office of Information and Regulatory Affairs approach information policy. Then, I would like to explain historical trends in computer security policy, relate that to the technical revolution we have had in the past ten years, and see if we can carry that one step further to see what new policy areas are going to be evolving.

Our office has viewed the "Whiz Kids" as good news and bad news. The good news is that we have gotten an awful lot of security awareness from their activities. The one thing everybody in this business knows, that we've been plugging for ten years, is that you have no security protection unless the people associated with information systems are aware of the need to protect them. In the Defense community, you've got very specific regulations and responsibilities for securing information built into everybody's duties. In the non-Defense community, there is not anything nearly as strict. So we are pleased about any security awareness we get in the non-DoD community.

The bad news is the perception created by what you see when you turn the TV on and watch the news broadcast. It is what the public perceives that we are doing with the information that we, as a government, collect from them - how we are protecting it. The government is an information machine and to the extent that we can't assure the protection of information we collect from the private sector, people are going to be more hesitant to give it to us. That relates to our tax data base, our census data, and all other data that we collect from the private sector. So there's a very real danger in bad publicity about security breaches inside the government.

My office is the Office of Information and Regulatory Affairs; some of you may remember we were in existence before the Paperwork Reduction Act of 1980. Since that Act passed, we have been in the Information Policy business. Under this administration my office is also leading the Federal Government's regulatory reform effort.

Our approach to Information Policy is two-fold: firstly, we are not in the business of regulating (we have little confidence in regulating the private sector - similarly the Federal Government), but we do have a responsibility to educate people who are in the position to make decisions; we need to be sure they are informed. Secondly, we have a responsibility to make sure that those decisionmakers have the proper incentives. We have to be sure that their incentives aren't skewed. One example of skewed incentives that we have often encountered in the computer business is the computer resource as a free good. Often the budget for a computer system is not that computer system program manager's budget responsibility. He doesn't have to defend it, so he views his computer systems support as a free good. And so he seeks such support even though it might not be cost-justified. That is an example of what we view as skewed incentives. We want to eliminate such distortions in information technology business.

When I discuss information security today, I mean non-national security. As I mentioned earlier, the national security community has a very long, very structured set of rules and regulations that most of you are familiar with. In non-national security information there are no such rules. However, we do need to protect such information. That is the kind of information security I am talking about today and that's the context of the policy evolution I will discuss.

Now I want to back up 15 or 20 years. Some of you will remember some of the technology of the 50's and early 60's. We had big pieces of iron that were single-threaded. What we had from a security standpoint was a Black Box; things went into the Black Box and came out of the Black Box. We were concerned with integrity, we were concerned with security of the computer room, but there wasn't much of an outside threat or vulnerability at that time. Unfortunately, that was also the time when the term "computer security" was first used. We have spent years trying to correct that misnomer - because computer security is really information security. Treating it as a technical Black Box issue is missing the whole point.

During the late 60's and early 70's, computer technology, as we all know, was evolving rapidly. There were big pieces of expensive hardware that were multi-threaded. There were multi-processors with remote terminal access. And more importantly, from an information security standpoint, we were automating more parts of information systems. Whereas, back in the 50's and 60's we used the computer to add up numbers in an accounting system, by the late 60's and early 70's, the accounting system was an automated system. Databases were automated and maintained on-line. It is important to keep the evolution of how more of each information system was being more automated in perspective.

By the later 70's, we saw the automated information security threat to the Federal Government as substantial enough to warrant issuance of Transmittal Memo #1 to Circular A-71. It was a multi-faceted approach to security of automated information systems. It talked about personnel security - about screening people with access to information systems; it talked about the traditional Black Box security - about risk analysis, computer room security, and contingency backup plans; but it also talked about a different concept, called applications security. It highlighted the importance of building security into computer applications, those "threads" that go through the Black Box. And talked about auditing those requirements later on. Underlying these three facets was a subtle shift in responsibility for security.

It was so subtle that I think most people missed it, but some responsibility for security in information systems was

shifting away from the Computer Center. It was shifting back to the Program Manager.

As we entered the 8Ø's, the hardware focus that we had in earlier years was rapidly fading. Hardware is only about a fifth of the cost of developing an automated information system now. Futhermore, the nature of the hardware has changed - we're into computer and communications networks. This presents a whole different problem from a security standpoint. A much more complex problem. Access to data networks, as we have all seen by the recent press releases, is virtually unlimited. To the extent we allow access to networks to equal access to information, we have little security over our information.

Today it seems the technology is moving even more rapidly. People are buying micros and end-user computing is at hand. But I don't need to tell you that the times are changing; you are all part of it. I understand most of you are in the technical end of our field. You're right on the cutting edge. So I'm not here to stand up and tell you about technological change. What I do want to communicate to you is that change may signal another shift in responsibility.

Back in the 4Ø's, before we had computers, line managers were responsible for their manual information systems. To take advantage of a developing computer technology in the 5Ø's and 6Ø's, we developed computer centers to assist the line manager in processing his information. With the processing change, some responsibility to protect the information being processed shifted to the computer centers while it was there. In end-user computing we are shifting technology into the hands of the line manager. With that change, responsibility logically shifts back with the technology to the program manager. But is he trained to handle it?

Recently, my office published a notice in the Federal Register announcing the development of a new information policy for the Federal government. Our intent is to consolidate four existing OMB Circulars into a single broad Federal policy for information management. Aspects of that directive will address how to cost information technology activities, privacy considerations in information management, and certainly the security of Federal information.

How does the Federal Government protect its databases? As end-user and on-line computing become commonplace that question has certainly become more difficult to answer. As our databases become more decentralized, how do we assure that information in those databases is properly protected? What role does the technologist play? The program manager? Those are questions I leave you with - I don't have the answers. In developing our information policy we are seeking the answers.

In closing let me reiterate that we want to hear from you. We want to develop policy that encourages the effective use of technology by the Federal Government. We can't do that in isolation from the technical community. So, I encourage all of you to participate in the development of that policy. Sometime this spring, look for a draft directive, and give us your thoughts.

A LEGISLATIVE PERSPECTIVE ON COMPUTER SECURITY

Tony Taylor

Congressional Committee on Science and Technology

Some of you may be wondering how the Science and Technology Committee comes at this issue. Generally, the Science Committee has authorization responsibility for non-military science programs in the government, including the space program, the energy program, the basic science programs of the National Bureau of Standards, and the National Science Foundation, the environmental programs, the transportation programs, and so forth. The one area which allows us to get into computer security is the responsibility we have for Communications Research & Development.

Generally, Congress has a very, very difficult time dealing with the question of computer security. The jurisdiction is split among many committees and for that reason Congress has a very difficult time dealing with the issue. For example, the Government Operations Committee deals with management of Federal Government information systems and computer systems, procurement regulations, oversight to central management agencies. The Judiciary Committee is going to deal with aspects of computer crime, privacy, and constitutional rights. The Science and Technology Committee, as I mentioned, has jurisdiction over the National Bureau of Standards, and there are a whole host of others who deal with specific agencies which have computer programs such as Treasury, Health and Human Services, Agriculture, and on and on.

Recently, the Science and Technology Committee held a series of hearings, as many of you are aware, and it came about this way: The Chairman called me into his office one day and said, "Tony, I was ordering some tickets for the Orioles game the other night and I did it on my phone, just by pushing a bunch of buttons," and he thought to himself that there must be a potential for abuse here. So he said, "Let's look into this and see if there isn't more to this issue," and sure enough, as we found out, there is a great deal more to it. The angle that we pursued initially was that we wanted to look at the Research and Development aspect of it, but we quickly found out there were many, many other issues in the regulatory, policy and management and legal areas.

We really were just able to scratch the surface in our first set of hearings, and I think the results were probably just a beginning to our investigations in this area. Some of the issues we looked at during this recent set of hearings were things such as the need for classifying non-national security information, both in the private sector and in the government sector. What is the nature and magnitude of the problem, and is it in fact a problem? Is the (Hacker) problem a serious one? Is the leadership by Office of Management and Budget and other central management agencies adequate in dealing with non-national security? Federal efforts in computer security technology and techniques by National Bureau of Standards and by the DoD Computer Security Center - are these addressing the needs of the government and the private sector? Is there a need for clarifying the law with respect to computer security? Is there a need for clarifying the law with respect to computer crime, and what actions should be taken by Congress, federal agencies, and the private sector who deal with any problems which may exist. I don't have answers for all of those for you at this moment; the Committee is still thrashing with these ideas, but there will be a report published probably some time next month, which will hopefully begin to come up with some of the findings in these areas. But, as I said, we'll probably have only scratched the surface and probably will have to pursue it again next year. In the meantime, we would welcome any input from all of you as we explore the issues further.

AN INDUSTRY VIEW OF THE DOD COMPUTER SECURITY CENTER PROGRAM

Robert H. Courtney, Jr.
Robert H. Courtney, Inc.

I am deeply obligated to both the Computer Security Center and to the NBS for the opportunity to talk with you today. Particular thanks are due those in the Computer Security Center with whom over the past few years I have established differences of opinion on several of the subjects to be discussed at this conference. I am truly appreciative of this clear demonstration of their objectivity through willingness to invite my somewhat contrary views.

First, I take no exception here today as to the appropriateness of the Computer Security Center (CSC) program to the needs of those groups within the DoD who find it necessary to store and process concurrently, and within the same or communicating systems, data potentially involving the national security and with more than one classification level; the classic multi-level security problem. I take exception today only to the reasonableness of Computer Security Center attempts to export their problem definitions and, consequently, their solutions to non-DoD agencies and to the private sector.

The CSC has no statutory basis for the expenditure of resources for the solution of computer security problems in the private sector or in non-DoD agencies. The CSC's interest and concern for the private sector security problems stem wholly from the hope that, if private industry can accept the CSC problem definitions, then these enterprises can use the security measures appropriate to those problems and which the DoD also needs. If that happens, the DoD can benefit from the economies of scale which would result from the large demand for those security measures. *These attempts to get lower costs are not unjustified; it is a quite sensible and well-conceived approach up to that point.*

Strong exception to the current CSC program is warranted, however, when it becomes clear that they have not committed even one person to understanding the needs of the private sector for control or security measures in and about computer-based systems while they continue their attempts to foster belief that the private sector problems are quite similar to the DoD multi-level problem.

I will stipulate here that CSC personnel have not and would not knowingly attempt to mislead non-DoD agencies and private enterprises about the nature of those groups' problems. The CSC assumption of similarity in the private sector problems and those which they postulate for the DoD multi-level environment has its orgin only in their lack of adequate familiarity with the security problems of most profit-making organizations. A majority of the DoD systems have security problems in common with the private sector; but this same majority of DoD systems have very few problems in common with that small but important portion of DoD systems which have the multi-level problems.

There can be no doubt, given the incredible diversity of applications made of computers throughout the business world, that there are some very few organizations which have problems roughly comparable to those seen by NSA in the DoD multi-level environment. But these are too few to constitute a market significantly large to be of help to NSA in getting better prices on those common security measures. To the extent that some defense contractors have the multi-level problem, they would have at least peripheral involvement with or interest in the CSC program, if for no reason other than the need to comply with contractually-imposed security procedures of the DoD - whether they are effective or not.

All of the attendees are now, or will be before this conference is over, familiar with the nature of the CSC program. The problem that NSA sees is, at least to a limited degree, implied by the measures under consideration there. Let's talk for just a few minutes about the comparable private sector problems.

We can look at 1293 cases of theft by computer in the three years ending last October 31. The vast majority of these required for their commission virtually no technical competence beyond the simple training required to operate a terminal in support of some normal business function, such as entering credits or debits into accounts receivable, making payroll adjustments, entering changes into accounts payable systems, or checking credit. Of the 1293 cases, only seven involved programmers, about Ø.5%. Only two of those seven modified the program for their own benefit; the others simply used the programs as they were intended to be used - but to perform an illicit transaction which they thought would benefit them.

The usual retort to my contention that we see almost no technical expertise applied to actual theft by computer of real economic consequence is that clever people do not get caught. In actual fact, there are very few companies of any size that are so poorly managed that they suffer major losses through manipulation of data and never notice. Theft requires that assets in some form be missing. They are not just missing from the company in general; they are missing from some part of the company, a cost center or a profit center, which is not going to measure up. They may be late in finding the loss, too late, in fact, to recover any of it, but they will not be forever ignorant of it. For that reason, I am convinced that there is no large number of such thefts resulting in significant losses which remain undetected. Losses show up on the bottom line.

As important as are the losses represented by these 1293 cases, they are quite small when compared with the losses to errors and omissions in those same organizations. Not only are mistakes very expensive, they are also the training ground for the potentially dishonest employee. These mistakes, not intrusions by James Bond's Russian cousin, Ivan Bond, are the principal data security problem almost everywhere and most particularly including federal agencies. Ranking immediately below them in relative importance, however, are the dishonest employees, many of

whom were able to observe in the course of doing their normal tasks that no means are in place for holding them accountable for what they do.

If means are not provided for holding all people having access to the data processing resource specifically and individually accountable for their actions, specifically including their mistakes, it will sooner or later occur to some of them that they can also make what may appear to be mistakes, but which, in reality, benefit those who make them.

When employees find that they are not held accountable for their mistakes, it then occurs to some of them, when their integrity gets stressed a bit by Christmas or a race track, a needy friend, or simple greed, that they can also make "mistakes" which benefit them. Such was the nature of many of our 1293 cases.

On the other hand, the losses in instances in which outsiders, without the involvement of insiders, penetrate business data processing systems to steal or illicitly modify data is quite small. If an outsider is involved at all in the theft or manipulation of data, there is almost always an insider, an employee, acting as an intermediary in the process. The hackers are exceptions, of course, but the economic impact of the losses, beyond those caused by adverse publicity, were quite small. We know of almost no instances of industrial espionage involving the theft of data from computer-based systems or from manual systems in which there was not also involved an employee authorized access to those data. *Most computer-related crime in government and business involves access to the data by people who were authorized access to those data.* This view of the problem does not align well with the CSC program.

Whether the introduction of the computer increased or decreased or had no effect on the amount of white collar crime cannot be determined. There seems to be a consensus among those familiar with both manual and computer-related economic crime that the nature of the people committing the crimes and their motivations have not changed greatly as a consequence of the introduction of automated recordkeeping systems.

It is quite clear that the introduction of the computer offered us an opportunity to greatly reduce the losses to theft and to mistakes. Our drive to implement new systems left little time to implement appropriate security measures and virtually no support for those who counseled caution. Because we did this, we denied ourselves many of the security enhancements potentially available with the introduction of computer-based systems.

In our 1293 cases, 26% were committed by female clerical employees, usually somewhat attractive, under 35 years old, induced by, and sometimes tutored by, a boy friend on the outside. This is not a comment on the relative trustworthiness of female as opposed to male employees. There is a clear predominance of women working in the clerical tasks. A surprising portion of these, however, did not see that there was much wrong in what they did, even when the losses to the organization exceeded a half million dollars.

Our data indicates that between 3% and 10% of the clerical, administrative, and lower level operational personnel will steal if they have an opportunity to do so and believe that they will not be caught. Again, however, these people are manipulating data to which they have access in the normal conduct of their assigned tasks.

The 3% applies to a fairly plush working environment, such as a major corporate headquarters. The 10% are more often found in very poor working conditions where the employees do not feel appreciated, where the physical environment is rather bad, and where employee loyalty is virtually non-existent. This does not necessarily argue for improving the work environment to approach the 3% level. That may cost more than is lost through theft. Remember, these are people who think that they will not be caught. Real economy lies in implementing enough security to assure them that there is a high probability that they will be caught if they steal.

About 11% of the people who are caught stealing from a computer-based system are referred to the criminal justice process. A large percentage of these are not then prosecuted. The result is that, of the 11% referred, about 18% are convicted. It is clear, then, that some 89% of those who are caught never face the criminal justice process at all.

Caution should be used in dealing with these figures. The use of percentages, such as 89, 11, 26, and 18, implies precision of measurement which is not possible. The gross relationships are quite correct. Whether 10, or 11, or 9 per cent of the cases go to prosecution is not readily knowable. Similarly, whether the portion lost to under-35 females is 26% or some other number slightly different is, again, not readily knowable. The score keeping is too difficult to let us be very precise. But then, such precision is not at all essential to the arguments made here.

What is most important in all of this is that the computer security problems faced by private industry impose costly losses, are technically unsophistocated, are, in the vast majority of cases, committed by people authorized access to the abused resource, and they are, in large measure, containable by simply holding persons using those systems rigorously accountable for what they have done.

People do not like to be caught making mistakes. They like even less being caught stealing.

Persons with histories of economic crimes rarely seek jobs in clerical roles involving the use of data processing systems in the hope that they might somehow find a way to steal. People inclined to theft usually have a more urgent need than that to satisfy their desire for easy gain. Thus, extensive background investigations to establish relative trustworthiness of people hired into most clerical roles involving the operation of computer terminals are rarely cost-justified. This comment is not necessarily applicable to people hired into other jobs, including, for example, bank tellers. I do not have data with which to make a judgement in those other areas.

The foregoing comments suggest that the computer security problems faced by private industry are not only much more mundane than those postulated for the DoD multi-level environment, they are also real, frequent and costly. To be acceptable, the measures which contain them must be available, highly effective, non-disruptive, and cost much less than simply tolerating the problems to which they are applicable.

The general failure of the private sector to adopt cryptographic protection on a broad scale when it became available should convey a very strong message to those in the CSC who look for support from private enterprise in generating a market for security measures which are intended to thwart far more sophisticated attacks than those currently being experienced. To the extent that some such orgainizations might be influenced by NSA's reputation for expertise in security matters to adopt very sophisticated measures not applicable to their real problems, NSA will have done them a significant disservice through causing a diversion of resource from the real problems to more esoteric, unreal ones.

Finally, industry experience with mail surveys or interviews in order to establish private sector interest in specific security measures has not been such as to encourage that approach to problem identification or to attempts to measure the acceptability of specific measures. In addition to the use of GUIDE, SHARE, COMMON, and other user groups for user input, considerable time spent developing a thorough understanding of the needs and attitudes and political environments in a diverse array of industry areas is essential to an adequate understanding of the applicability of specific security measures to those particular environments. There is no possibility of extrapolation from the DoD multi-level environment to gain an understanding of the private sector security needs. Because that is true, it is then unreasonable for the NSA Computer Security Center management, which has not conducted an adequate study of private enterprise security needs, to claim relevance of their program to the needs of any significant portion of private industry.

PANEL SESSION -- BASE SPECTRUM OF COMPUTER SECURITY REQUIREMENTS

Moderator - Dr. Stuart Katzke

Manager, Computer Security Management and Evaluation Group

Institute for Computer Science and Technology, NBS

Panel Members:

James P. Anderson - J.P. Anderson, and Co. (No text available)

William H. Murray - IBM,Inc.

Nancy Woolsey - Lockheed

Jimmie E. Haines - Boeing Computer Service Co.

INTRODUCTION

The primary focus of this session is to try and identify security requirements which are common to both the Department of Defense and the non-DoD community; that is, the private industry community and the part of the Federal Government which is non-DoD.

It seems that over the years, there has been a difference in requirements between the Department of Defense and non-DoD communities. This perception, I think, exists primarily because of the DoD Computer Security Policy. Most DoD organizations don' t really question whether the DoD policies are applicable to how they classify and handle data.

When you look through the DoD Trusted Computer System Evaluation Criteria *you can' t help but notice requirements which include things like user identification or authentication, audit, system integrity, security testing, design and test documentation, sensitivity labels or methods of classifying objects, design specification and verification, user' s guides to security features for systems, and configuration management concepts. I think we would all agree these concerns are common to the entire user community. On the other hand, when you look at the Criteria, you also see requirements that are designed to take care of specific DoD needs.*

So I think one of the things we would like to try to do in this session is to identify those common requirements. I' ve asked the session participants to present their perspectives on basic or fundamental requirements based on their experience within their organizations. As a part of their response, they were asked to consider the applicability of the DoD Trusted Computer System Evaluation Criteria to the non-DoD community.

William H. Murray

INTRODUCTION

My remarks will contrast the security requirements of private sector organizations to those of the DoD as reflected in the Trusted Computer System Evaluation Criteria. Several comments are required to set the context. First, it is my personal belief that little useful can be said about the security of a system except in the context of a particular application and environment. Because of the differences between the applications and environments of the defense and private sectors, it should come as no surprise that their security requirements are different. Second, I will contrast requirements; i.e., I will compare the requirements of the private sector to those reflected in the Criteria. Therefore, I will probably not completely cover the requirements. Since the task set for me is to draw distinctions, I run some risk of sounding critical of the Criteria. That is not my intent. I believe that the criteria represent a valid reflection of the DoD requirements. Third, while I will not comment specifically on the requirements of the non-defense public agencies, I believe their requirements to be somewhere between those of Defense and the private sector. Finally, while I believe that there are substantive and substantial differences between the security requirements of the DoD and those of the private sector, I still feel that objective criteria such as these can be useful in both areas.

CLASSIFICATION AND CONTROL PROCEDURES

The criteria assume a set of security objectives for DoD. These objectives are reflected both in the security procedures followed by DoD and in the security policy required by the TCB by the criteria. While it is possible to map the private sector requirements with those of DoD there are important differences in environment and emphasis. For example, much of the function required in category B is required to support the DoD's data classification and control procedures. The number of private sector organizations with such a system of classification and control can be enumerated on your fingers. Therefore, few private sector organizations can avail themselves of the functions required in category B.

NON-DISCRETIONARY CONTROL

The Criteria reflect the DoD's requirement for what are termed non-discretionary controls. The concept is based upon the fact that these controls have their justification in

legislation and regulation. There is no such concept in the private sector. There, all controls are discretionary. The discretion may be reserved to some designated function or level of management, but they are still discretionary. Even in DoD, there is some function or level of command that has discretion over those controls. Non-discretionary really means discretion reserved from the owners and users of the data, and to the commander and the security staff who grant security clearances.

In the private sector there is no analog to a "clearance." Therefore, in the private sector, even those organizations that classify data, still grant access only on the basis of need-to-know.

CONTROLS AGAINST DOWNGRADING

Part of the B requirements in the Criteria relate to maintaining the "star-property," i.e., controls against downgrading of classification. In Dod even the person who classified the data in the first place does not usually have the discretion to lower the classification. In the private sector, the classification is a communication from the author of the data to holders in due course about how the author believes that it should be handled. However, it is a judgment of the classifier, rather than a property of the data, and it clearly does not have the force of law. The discretion to classify includes the discretion to re-classify, up or down.

For example, the classification rules that are used in IBM say that a copy of a data object should be classified the same as the original. If the output of a job contains essentially the same data as the input, it should be classified the same as the input but that if it is a substantive transform of the input, then it should be classified on its own merit. In either case, it is the responsibility of the owner of the copy to see that it is properly classified.

MEDIATION VS. ISOLATION

It's also because of the requirement to maintain the star-property that DoD must rely upon the operating system for mediation of data sharing as well as for isolation. Since this is not a requirement in the private sector, the operating system is relied upon primarily for isolation, that is, protecting the application and its data from any outside interference or tampering. Mediation of sharing is often done in a data base manager or in an application subsystem.

CONFIDENTIALITY VS. INTEGRITY

Indeed, in the private sector, perhaps as little as one percent of the data is sensitive to disclosure, but most of it is sensitive to modification. Thus the emphasis is on data integrity rather than confidentiality. While the kind of access control indicated in the Criteria is sufficient for confidentiality, it is necessary but not sufficient for integrity. Therefore, in the private sector, there is more security emphasis on the application than in DoD, and less on the operating system.

In that class of events as "computer crime" but which my colleague, Stan Kurzban, describes more accurately as "crimes which the proper use of computers could have prevented," it has invariably been application rather than operating system control which has proved to be deficient. In most cases, the entire effect has occurred totally within an application where it was both invisible to and immune from the operating system.

ACCESS CONTROL VS. AUDIT TRAIL

The evaluation criteria place relative emphasis on access control versus audit trail. This reflects in part the relative difference in the requirement for confidentiality versus integrity already noted. In part it also reflects the difference in vulnerability to unauthorized parties versus authorized. In defense applications, one must assume a determined, resourceful, outside adversary against whom audit trail is ineffective for either deterrent or for remedial purposes. However, in the private sector, the exposure is to error and crimes of opportunity by authorized personnel. The role of access control here is as a necessary condition to the integrity of the audit trail. However, it is the audit trail that offers both the deterrent and the remedial effect.

ACCESS CONTROL VS. USER IDENTIFICATION AND AUTHENTICATION

Similarly, the Criteria places emphasis on the consistent application of access rules. Effective and efficient user indentification and authentication are assumed. In the private sector, problems with unauthorized users have centered on unauthorized access to the system rather than to data. Therefore, the requirement is for effective and efficient user identification and authentication, i.e., for access control to the system.

DEMONSTRABILITY VS. WARRANTY

Because of its high reliance on the effectiveness of the operating system, the defense establishment places high emphasis on the ability to test or demonstrate. Because of the demands placed on its own knowledge and resources by such demonstrations, they are delegated to a third party designated by appropriate authority. Purchasing authorities are permitted to rely upon the judgements of the third party. Evaluation costs can therefore be spread across a number of units.

In the private sector, representations of disinterested third parties have less value even than those of vendors. Public representations made by vendors to large numbers of customers, even with limited remedies, are more useful. The individual customer gains confidence from the wide exposure to discovery by others that any flaw would have. This is analogous to the way he gains confidence in tables of logarithms, for example.

Demonstrability is a property of an operating system that is achieved only at the cost of other desirable properties such as performance, function, generality, flexibility, longevity and extendibility. The more systems instances you can spread it over, the more likely you will be willing to pay the price. Thus, with its large population of systems, DoD is more interested than the small entrepreneur and indeed more than the commander of a small unit.

ENFORCEMENT VS. ARTICULATION

The Criteria is strong on the effectiveness of the system in enforcing access rules, but is relatively silent on the support to the commander or manager for articulating them. This reflects in part the fact that in DoD the rules are habitually understood and are relatively stable, and the assumption that those setting up the rules are knowledge-

able, conscientious, and motiviated by harsh, legal sanctions.

Experience in the private sector suggests that far more damage results from the rules themselves than from any failure to enforce them. Either they do not reflect the policy or good practice, or they have not been updated to reflect changes in the environment, such as changes in organization or responsibilities. Therefore, requirements and selection criteria for the private sector would have to give far more weight to services such as alarms for alerting management differences between the rules and the situation, and services for facilitating the creation and maintenance of the rules database.

CENTRALIZED VS. DECENTRALIZED CONTROL

Finally, the Criteria seems to envision a single system-wide integrated process which acts as a surrogate for all management and with all privileges reserved either to the systems managers or the security staff.

In much of the private sector today, the emphasis is on networks of distributed systems rather than single homologous systems. In such distributed systems, no single process has effect in all parts of the system. Security must therefore depend upon contention and cooperation among subsystems, rather than on the correct behavior of any single subsystem.

Indeed, security based upon the correct behavior of a single system assumes that the objectives of the management of the system are homogeneous. While this may be true in a hierarchically organized institution, such as the defense department, it is not true in the private sector. Much of the private sector is organized in contending lines and staffs or in networks. Such organizations work specifically because all authority and privilege never become centralized. In many of these organizations, line managers would not tolerate the centralization of discrection or control over their data in the hands of the system or security staffs. This is analogous to the concept employed in weapons systems, where an individual is trusted not because he has passed the background investigation or because his scope is narrow, as for example in compartmentalized intelligence, but because he checks and is checked upon by others. Thus, the evaluation criteria for the private sector would have to reflect the requirement for controls which are contending and exclusive rather than hierarchical. An example of such controls appears in the MVS/RACF where operations personnel may have privileged access to the data, but cannot alter the access rules; security administrators have privileged access to the rules but not to the data; neither has control over the audit trail since that is reserved to the auditors.

CONCLUSION

These are some of the differences in requirements between the defense sector and the private sector that are brought to mind by a review of the Trusted Computer Base Evaluation Criteria. Evaluation criteria used by the private sector should reflect these differences.

One parting comment about the criteria themselves. The hierarchical ordering of classes appears to imply a relative goodness more than a difference in the requirements to which they respond. I don't know whether this is intentional or accidental but I am concerned it may lead to some mistakes. I expect any day now to see an RFP for an operating system that requires all the generality, flexibility, and richness of an MVS and that is also an "A" system. I believe that to be an over-constrained set of requirements.

I am also concerned that this ordering of classes suggests that demonstrability is more valuable than implementation of the DoD policy is more valuable than function is more valuable than effectiveness or efficiency. For example, because it only knows about DASD extents rather than data objects, VM can only aspire to satisfy the D criteria. MVS which is both richer in function and bigger in size can expect to be evaluated as a C and might even aspire to be a B. On the other hand, because of its safe defaults and limitied size, in some applications and evironments VM can be more effective, efficient and demonstrable than MVS.

Nancy Woolsey

I have experience, both on the DoD side and on the industry side with a company that's 98% military oriented, but has a substantial amount of unclassified processing in that area. As a representative of such an industry, I hope that to a certain extent I am also speaking for an audience that, for the most part, is absent today: that is, the audience that is composed of purely commercial concerns, that have no direct involvement with the government, and by and large, never have had.

Over the past seven or eight years since I have been directly involved with computer security (and I admittedly am no pioneer in the field), I have been to a number of conferences sponsored both by the military or other government agencies and by private interests, and at every one of them the subject of "typical requirements" has been discussed. Nevertheless, it's being discussed today. The subject was discussed here last year by several speakers both from the government and from commercial concerns. Now, I certainly don't object to discussing these baseline requirements, but I have to wonder just a little bit why it is that after so many years we are still at the point where we feel we need to keep conceptualizing that which we are supposed to be applying to our various disciplines.

I would like to discuss this morning some general principles concerning those typical requirements and then maybe take them one step beyond, to what I believe should be the major focus of our interactions as information security specialists in the future.

I believe that I have a pretty good conceptual picture concerning the information security protection requirements in my organization. I hope so, since I'm the one that's putting them on paper and implementing them in specific situations, and the elements that I have to look at to assure to the best of my ability that the information vital to my company's interest, both commercial and military, is being adequately protected. Now, to achieve that standard of accuracy, that which I define as my baseline requirement for security, I have to look at a number of characteristics which are involved in a decision as to the security integrity, the total security integrity, of a given system or network, and these go beyond the items that Stu enumerated this morning. That's one aspect, but there are others. The

elements include (a) the physical security system; the integrity of the hardware, the areas in which the systems or connections between systems are located, and how they are located; (b) personnel security: the trustworthiness of the people that we hire, based upon the standards that either we or the government, as appropriate, have decided to enforce; (c) software and data security: the extent to which our information, including hard copies and output, is protected in accordance with established policy, including assuring that the implementation of the security software package that we might use is to design specifications and that access control mechanisms are being rigidly enforced within the constraints imposed by the capability of those packages. I think that this function, assisting in identifying and controlling the vast amount of information processed by and stored in our system is the single most complex, challenging and time consuming aspect of our function.

Fourthly, network security: that the various physical, personnel and access controls that we have established for our mainframe systems are maintained in a satisfactory manner throughout both our local networks and our teleprocessing networks. And, lastly, the policies and procedures that we have developed must reflect the current thinking of our company, be in accordance with what actually exists, and implemented in accordance with those policies. Moreover, these procedures must address the system throughout the processing life cycle and must be understood and accepted by those who are responsible and entrusted, to implement them on a day-to-day basis. This involves routine and frequent follow-up, investigation of abnormal conditions or security violations, and appropriate reports to management.

There are, of course, other issues which can be included in a typical baseline requirement for security, for instance, disaster recovery, contingency planning and related types of activity. Now, some of my colleagues are responsible for these functions. I'm not. But to the extent that others are so responsible, they do have to be included in defining the baseline requirements of security.

Now, I have not defined specifics within each of the categories that I mentioned before. We could give you 2Ø questions to ask (or not ask) a prospective employee, 3Ø things to look for when constructing an area or describe our two profoundly different implementations of ACF2. But, these are all in books and articles in the literature, and they are available. The point is, I know what my company's minimum requirements are, but even within my own company, the implementation of those requirements changes from system to system, as the nature of the system, its inherent controls, and its intended usage changes. Every implementation procedure which I have produced has been different from every other to one degree or another.

Hence, my concern for defining baseline, or "typical" requirements. I'm not certain that we haven't defined these requirements already to the best of our ability. In talking to my colleagues in other commercial industries, banking, insurance, retail and the like, even those totally removed from any contractual involvement with the government or military, I find we still talk a common language. We look at pretty much the same thing, but we look at them differently. This is the difference I would like to concentrate on: the perception. I submit that what is becoming, and properly so, an increasingly important issue, is not only defining the nebula of baseline requirements. As our field matures, we need to define and understand each other's flexibility requirements within that baseline, because although the relative components of typical security requirements may not change, the relative priorities do. It's not enough to say that government and industry must work together, must team up, must share notes, must cooperate; we must define common interests before we can define common parameters. Maybe, that common interest does not exist to the extent that we would like to make it exist. After all, you don't draw a single blueprint and expect to build every house from it. Some houses require earthquake fortification and some need slanted roofs to allow snow to melt off.

Therefore, rather than talk further about the definition of typical baseline security requirements, I'd like to focus a little on industry problems or needs, which in turn drive or impede our implementation of security requirements within our industries based on our separate analyses. I believe that defining *those* needs and seeking a team approach to finding a solution to our problem areas is where we should be focusing.

Industries, across the board, face some common impediments to development and installation of adequate information systems security programs. I think the five most common problem areas are: the money problems which industries face; the man-power problem; *what* to protect; the definition and valuation of information in relation to other items in a company's inventory; and the ever-present management "problem" - as we perceive it, not necessarily as management perceives it. Problems which dovetail into these major categories or contribute to the basic problem are a lack of understanding of the problem itself; the problem of allocating scarce resources; the problem, which I defined previously, of ordering our priorities; and instituting security controls - retroactively in many cases, and that's a practical fact of life, for we are all slaves to our systems.

Security is expensive. Information security is really an overhead function that supports an overhead function. Moreover, our success is basically unquantifiable. None of us is going to get a blank check, except under the most extraordinary of situations. And, frankly, I don't want one of those extraordinary situations in my company. I don't want banks to supplement my paycheck, either.

I was talking to a friend of mine the other day who is an employee of a major, and healthy, bank which does business in the western United States, and he surprised me by stating that in the banking industry, a one percent net profit annually is considered a good profit. Now when you're talking about ratios in that area, you're talking about very conservative internal spending. Furthermore, not only are you not likely to have much to in this environment, but what you spend your money on is going to be very carefully scrutinized. Management, as a practical matter, is not going to let you go out and spend a goodly sum of money on a network message validation security package when the installation of a state-of-the-art ethernet will earn more profit from EFTS transactions. This means, of course, that in this type of industry as well as similar industries, the information security officer has to

be creative as well as cost-effective, and has to know how to sell his most basic requirements. The unique lessons learned in this type of environment can, and should, be shared with others.

Back to manpower. We all have a problem in this area, or a variation of it. If we have enough slots in our organization to cover everything that we think should be covered, we can't find the people to fill them. There is tremendous interaction in the industrial world (interaction is the polite word) aimed at finding people to fill positions. However, if you have that problem, you're lucky. The majority of us suffer from the opposite problem - too few slots as well as too few people. Management in general has not been given a reason to become that concerned about the ramifications of a loss of security to attach the same importance to having what *we* might consider an adequate, well-staffed, security organization. In their behalf, I might say that we as practioners are partly at fault. Management is understandably reluctant to acquiesce to that which it doesn't understand, and we are sometimes hard put to articulate our concerns intelligibly. Note: I did not say intelligently. Faced, therefore, with the problem of lack of manpower, we as security specialists are forced to prioritize our requirements, sometimes in an unhealthy way. For instance, I am charged with both implementation and inspection responsibility, but I can't do both, with 4Ø systems, some large, some medium, some small, up and running, and more being installed all the time, plus changes to the existing systems. I have to consider implementation of security rules and education of the people charged with that implementation as having first priority. I'm lucky if I actually inspect a given system once every six months. Most security violations I leave up to internal data processing personnel, my security monitors, to investigate and cleanup. Moreover, I have no direct authority over those people.

To try and overcome this problem adequately. I submitted to management recently a concept for a security team approach which would include some non-traditional positions, including programmer analysts, data base administrators, systems analysts and miscellaneous users, both local and remote.

In doing so, I have found it necessary to define certain items: (a) the structuring of teams and the provision of responsibility and authority to them; (b) the principle of evaluating the team's effectiveness rather than the individual's effectiveness in enforcing guidelines and standards implemented through the conceptual requirement for local review and feedback on those reviews; (c) involving the team in on-going education, both of themselves and of the people for whom they are responsible.

I believe that a teamwork approach, if presented properly to management, can be effective in many organizations. It need not cost a lot of money, and not necessarily a great deal of time on the part of the team members. It does involve instilling a continuing awareness of security on the part of the team members, which they will - hopefully - pass on to others within their organization. It's only a panacea, but it does, I think, partially solve the manpower problem.

The third problem revolves around a decision, which ultimately has to be made by management, concerning "what to protect." What is more important? Protecting your programs and data by suitable software means, or effecting the rapid transfer of information necessary to transact business? Is the integrity, the security, of your network as important as the accuracy of data transmitted over it? Can the security you feel you need be handled on a cost-effective basis? The answer always has to define what is necessary, not what is ideal.

Does everything have to be protected? This is really the most basic issue and the one that dovetails most closely back to the issue of defining baseline security requirements. Do we concentrate on hardware or software, users or access, data or output? These are all decisions each company has to make, based on its own business, its own profit-making activities and its own perceived unique threats. And, you do have to have someone who can define the threats as well as the priorities attached to thwarting each threat.

The fourth problem area industries face is the problem of definition concerning the sensitivity and valuation of their information resources. Here I'm using the word information in a narrow sense. A number of private industries have classification or designation systems designed to place a sensitivity label on their information, for instance, or restrict access in some other manner. But then they leave it up to the individual to do it, with little or no centralized review of that labeling. Now, in some industries, the designation system is archaic, not fitted to the complexity of current needs. I am aware that in other companies pitched battles are raging concerning the definition of an owner. Does the same person who has the right to create and access the data in all cases have full control over it? Does this imply responsibility, and if so, why isn't that stated?

I bring this problem up because I believe it is of particular concern within our area of responsibility. We need to be able to designate that which is vital, that which is merely essential, that which is important, that which is unimportant, that which is transient and that which has to be retained. And, we need to be able to review those decisions made on the basis of the classifications scheme that we develop.

This area is also of particular concern within industry for an additional reason. Computer security technologies are proliferating and they are exciting, many of them: encryption equipment, call-back mechanisms, validation schemes, software security, our discussions here concerning trusted operating systems. But, the fundamental question for industry is not only how do you use these new tools of technology to maintain security and still attain productivity, but also, how does industry put itself in a position to use the technology? If we can't classify our data, and maintain control of our personnel, how can we even attempt to justify our acquisition of security technology? How could we use it if we *did* get it?

I agree in principle that the concepts embodied in the DoD Trusted Systems Criteria can be applicable in theory to many industries across the board. But the fact is that application of these criteria implies a good internal classification system pre-existing, if it's going to be used on a practical basis. Frankly, much of industry is in disarray in this area.

To go back to square one and classify everything in our systems would be extremely complicated, time-consuming and expensive, and I question whether a piecemeal fit, as I have heard advocated by a few people, would really accomplish the objective. I'm not sure whether it's too early or too late for the application of such systems but I'm not sure it's the right time. I hope it's too early.

I'm also going to throw something else out to you, particularly you who are involved with the DoD Computer Security Center and the ICST. If you want to make this Criteria applicable to industry, you're going to have to identify the industrial community that you are really interested in reaching, because they are not going to come to you.

The last problem area I have identified is the problem of management awareness. We've talked this one to death and it still keeps raising its head. One reason is that we have to realize that if our management is aware of the threat as we present it, they still have to balance that threat against loss of revenue by other means. I include here a prevalent management perception that imposition of security controls will slow things down or will impede productivity. And, in many cases, they're right. If we are going to make a difference, we have to be prepared to discuss these issues in a positive manner. It may not be that management is not listening, it may just be that we aren't saying it right.

In summary, it is my perception that the general malaise that a lot of us suffer from is caused not so much from a lack of understanding and definition of our typical baseline requirements for protecting our companies' information resources, but our lack of knowledge concerning our mutual problems. As the profession matures, we have to focus on our diversities rather than on our similarities, and yet recognize those problems which do cut across industries and will always be with us. It is through discussion of our mutual problems that our diverse but ultimately similar though not identical security requirements will find a foundation upon which to build. We must participate industry-wide in developing and recommending solutions to those problems by sharing experiences, methods and results.

Jimmie E. Haines

I'd like to approach this presentation a little differently. First I will give you a little background. For those of you who are not familiar with the Boeing Company, if you flew in to attend this conference, it's more than likely you flew in on a Boeing airplane. If you watched the news last night, they were discussing the deployment of the Cruise Missile in Europe: Boeing built the Cruise Missile. If you are associated with some of the top five companies in this country, more than likely, you've used some of the services provided by Boeing Computer Services (BCS). The reason I bring this up is to give you a little perspective on the involvement we have in computing in BCS. We are the division of the Boeing Company providing all computing to the company as well as computing products and services to the commercial marketplace. BCS products and services include business systems, technical systems, scientific systems, consulting, education and training, as well as those systems servicing DoD requirements.

You've heard from the two previous speakers about the data protection policies and procedures, their implementation, and their maintenance. I would suggest that you need to make sure that your data protection policies and procedures are invoked within the operating components of the organization.

The contrast between DoD and the commercial sector is fairly significant. If you look at the requirements that are identified in the orange book, the requirements for the protection and segregation of the categories of data are clearly stated for the different levels that one might view in approaching the classification problem. Since our involvement at BCS has been with DoD for many years, we think we fairly well understand what those requirements are, and we believe that the orange book identifies the requirements related to trusted systems levels fairly well.

Constraints exist today in the computing systems environment that are imposed by the multi-levels of the Federal Government and DoD security areas which do not have common requirements. There are still significant difficulties to overcome having to do with some of these constraints, such as whether you're able to process in a multi-processing environment. Questions remain as to whether or not government and DoD requirements are still valid for completely stand-alone, segregated, completely encapsulated processing environments. I believe there is a significant challenge to be addressed by industry as well as the Federal Government to address these types of requirements.

The classifications that are noted in the orange book are representative of the kinds of tiering that are needed so that one can understand what the different sets of privileges should be and, more importantly, what should be accommodated in policy and procedural direction and in its implementation within any operating organization. The cost drivers that are associated with classification and multi-level systems and with their implementation and measurements certainly have to be considered. The cost of development and implementation has a great bearing on how an organization may approach its ability to implement the higher level security systems.

There is no doubt in my mind that as the technology explosion continues, and we get heavily into architected networks and distributed processing, it will become more important for the problems associated with multi-level security to be solved.

As Bill mentioned earlier, there are differences in the requirements for data protection. As I noted earlier, these differences are not necessarily generated by the contractor, but respond to DoD requirements that are specifically stated in contracts.

On the commercial side of the operation, it is left up to the companies to understand their data protection requirements and, more importantly, the data they are protecting. If you contrast the requirements that are identified in the orange book from a DoD perspective versus a commercial perspective, there are similarities that exist and that should be considered within an operating organization that deals principally with the commercial

kinds of activities. In our business we have been involved in both commercial and DoD activities and have developed a strategy in the corporation to have policies and procedures, and, more importantly, the mechanisms for their implementation.

I believe these similarities that exist between the requirements in the orange book defining DoD security levels and what companies should consider can be viewed as sound business requirements for protecting computing environments. The companies involved in commercial operations must clearly understand what the levels and differences are that need to be addressed on issues of proper assessment of the data, its worth to the operation, and the value of protection necessary in regard to the cost of whatever protection features you want to provide.

In viewing the emerging technology, we are getting into significant distribution of processing and are in the position of having available to us significant computing power, particularly personal computers. I believe that the question of data worth is becoming more and more important to us. That being the case, I think it will drive the necessity for proper classification of that data. Also, I believe it will drive individual companies to reassess the positions they have had in the past on what kind of a schema they really want to use. There is no doubt in my mind that there are levels of protection that are mandatory for the commercial business sector in regard to the value of the data as there are in the military sector for national security kinds of requirements. What you're really concerned with, I believe, is the granularity of protection capabilities, and I believe you're also very vitally concerned with the impact of a loss, either from an integrity point of view or from the divulging of sensitive information relative to the company operations.

One of the more important things that needs to be addressed is continued training and education starting at the highest level in the company down to those people who have to implement the various policies, procedures, and data protection capabilities. It is easy to write policies and procedures and to release them throughout an organization. It is not quite so easy to ensure that they are implemented. It is even more difficult to ensure that they are implemented properly. It is my contention that perhaps the discussion of data security itself has generated a tremendous mass of information that is a little difficult to digest.

What I believe to be fundamentally important is for each operating company to assess the value of its data in respect to its operations, and then to accomodate that with necessary policy, procedures and implementation instructions, and to follow that up with proper auditing to ensure that it was implemented properly.

Now, I'm switching sides for just a moment. From the data processing perspective, the one thing I believe is very important is that there should be an on-going research effort to analyze the emerging computing technology and the kinds of safeguards that you can offer to the community using your services. I believe that it goes without saying that you could have the best technological safeguards that you could possibly dream up, but if they're not being utilized within the organizations they are intended for, they are of little use.

With the kinds of activities in Boeing Computing Services, we are addressing all sides of this issue, and I believe it is important for us to continue to address the emerging technology and to provide sufficient technological investigations or analysis of those products that may be emerging from the vendors and to assess them against our own internal needs. I think it goes without saying, and I'm sure that everyone in this audience recognizes that it is so, that if you compare the levels of requirements for data protection illustrated in the requirements of the orange book versus those that are internal to a company, you'll find some similarities, perhaps, not to the granularity that the orange book specifies, but to some degree relative to classification. Consequently, the implementation of computing hardware and software protection safeguards throughout industry varies from very little to maximum protection capability.

In our organization, we have implemented provisions for processing DoD classified data and a system for internal company critical systems classifications, associated processes, and a classification schema for data. What we attempted to do in the recent past was to ensure that our safeguards in computing are appropriate to protect information to the levels of protection that are levied on us by our customers and the company using our computing services.

I think if I were to summarize my thoughts on the differences between the military as a DoD, non-DoD relationship that Stu had asked us to look at, I would believe that there are certain requirements that are analogous to what the orange book has in it that need to be carefully considered for application within the commercial business sector. I believe there needs to be a classification schema of data, and once that has been decided, there has to be commitment from the top levels in the company down through the implementing organizations and that it has to be implemented in some way in which those people who have to physically do the work can understand what is required.

In evaluating an organization's understanding of who is responsible for data protection, if you asked about the classification process once you had implemented it and the answer came back, "What process?" it would certainly be an indication that you might have failed. I think it goes without saying that each member of an organization is responsible for data security which drives the fundamental requirement that there be proper instruction/education provided throughout the organizations.

It may just be that we have sufficient requirements identified, but that we haven't communicated them to everyone who is concerned with the protection of data; and consequently, we suffer the problems that you have seen recently. In our company, we have a system that doesn't solve all the problems, but it's a foundation. It is a baseline to work from, with which we have had some success. I think that's where it all has to start. It would seem to me that we have to really look at the point that was made earlier by Nancy, that you have to view this question in terms of your own organization, and take appropriate action to ensure that your data and computing resource protection requirements and capabilities are responsive to the company's philosophy, the value they've placed upon

data, and that the corresponding risks your company is willing to take are properly addressed. Thank you.

TRUSTED COMPUTER SYSTEM EVALUATION CRITERIA: MAJOR DIVISIONS

Sheila Brand

DoD Computer Security Center

My name is Sheila Brand. As Chief of Computer Security Standards at the DoD Computer Security Center, I was responsible for the development and publication of the DoD Trusted Computer System Evaluation Criteria. This morning was extremely interesting as many of the speakers expressed their thoughts on the applicability of the Criteria to private sector needs. I don't know how many of you are familiar with the Criteria, and so, I will begin by giving you an outline, in layman language, of what the Criteria is all about. This entire session is dovoted to a discussion of the Criteria, and not to its relationship to the non-DoD part of the government, or to the private sector. We will concentrate strictly on features described in the Criteria. Hopefully though, if time permits, I would also like to address some of the issues raised this morning.

The Criteria is a document that has been developed over a five-year period by a number of very talented people. It has undergone extensive review. These reviews have resulted in a number of revisions and iterations of the first version prepared by Grace Hammond Nibaldi. And I think that the effort has yielded significant results. The Criteria represents an objective and accurate effort at articulating the technical issues of the computer security problem. Solutions to problems that are presented are solutions that we in the Department of Defense feel are needed. Whether they are needed by the private sector is not the focal point of this session.

The Criteria is a technical document; it is not a procedural document. The Criteria addresses hardware and software security features, their implementation, and problems associated with implementation. It also addresses the security requirements of the development cycle: configuration control, and the management and techniques used in secure software development. The Criteria was originally developed for a number of reasons. First of all, we wanted a basis, a yardstick, a metric, if you will, with which to measure the amount of security present in a specific computer system which was to be used for the secure processing of classified or other sensitive information. Second, we wanted to provide guidance to manufacturers as to what security features to build into their new and planned products. And third, we wanted a method for uniformly specifying security requirements in acquisition specifications. Given this diverse set of needs, development of the Criteria became an extremely complex task. It may be the first document of its kind, in terms of trying to articulate in rather simple terms a large number of requirements.

The Criteria is divided into four hierarchical divisions. Divisions are subdivided into classes. Within each class we have arranged the features according to control objectives that each feature satisfies. There are three major control objectives against which the Criteria was developed. They deal with security policy, accountability, and assurance.

The first control objective deals with security policy. The Control Objective says:

> "A statement of intent with regard to control over access to and dissemination of information, to be known as the security policy, must be precisely defined and implemented for each system that is used to process sensitive information. The security policy must accurately reflect the laws, regulations, and general policies from which it is derived."

In other words, before a set of security measures can be designed and implemented in a hardware/software system, the designers should have a clear sense of the perceived threats, risks, and goals of the organization for which these measures are being developed. This perception is often articulated in terms of a security policy. Furthermore, this policy should accurately reflect the laws, regulations, and general policies from which it is derived.

In addition, there are three sub-objectives to security policy. They deal with discretionary or need-to-know access control requirements, mandatory access control requirements, and sensitivity labeling of data.

The second control objective deals with accountability. It says:

> "Systems that are used to process or handle classified or other sensitive information must assure individual accountability, the capability must exist for an authorized and competent agent to access and evaluate accountability information by a secure means, within a reasonable amount of time, and without undue difficulty."

Requirements arising from this control objective include: individual accountability for actions in the system, positive identification and accurate and reliable authentication of that identity, plus the ability to record, maintain and use audit trail information of security-related events.

The third control objective deals with assurance. It states:

> "Systems that are used to process or handle classified or other sensitive information must be designed to guarantee correct and accurate interpretation of the security policy and must not distort the intent of that policy. Assurance must not distort the intent of that policy. Assurance must be provided that correct implementation and operation of the policy exists throughout the system's life-cycle."

In other words, assurance is concerned with providing guarantees, throughout the system life cycle, that the system of controls that is developed accurately implements the security policy set down by management, that those controls provide the needed accountability, and that when

in operation, the system will, in fact, carry out these control functions and these functions will not be abridged.

The remainder of this session deals with the divisions of the Criteria. I will discuss the Division C and the Division D. I will leave it to my colleagues at the table to discuss Division A and Division B. Dan Edwards, Chief of the Office of Standards and Products at the Computer Security Center, will discuss Division B systems. Dr, Carl Landwehr of the Naval Research Laboratory will discuss Division A systems.

The first division of the Criteria, Division D, deals with what we call minimal protection. Users of Division D systems do not rely on hardware and software controls to protect their information; instead, they rely on procedural controls, administrative controls, and physical controls. We do not, at the Center, evaluate Division D systems. I doubt whether any vendor would come in and ask us to do so. There are no classes in Division D nor are there any features.

Division C is the first division which delineates requirements. Earlier speakers described some very serious problems of computer crime and abuse in the private sector. I believe that the controls that are enumerated under Division C of the Criteria are just the kind of controls that could have deterred these crimes. Division C basically provides accountability.

Features required of Class C1 include need-to-know access control between named users and named objects (e.g., files) of the system. Also required are user identification, and authentication of the user's identity. As a point of reference RACF, ACF2, and Top Secret security packages provide features delineated for C1 and, are all being evaluated by the Center.

In addition to the features required of C1, the following are required to meet Class C2 requirements: (1) Need-to-know controls must provide the capability of including or excluding access to the granularity of a single user. (2) The notion of object reuse is introduced. This control prevents a user from scavenging through either memory or storage to obtain information for which authorization has not been given. (3) Individual accountability is required. That is, the system must be able to uniquely identify each individual system user and this identity must be associated with all auditable actions taken by that individual. (4) Audit trails are required. That is, the system must be able to record, in an audit trail, a series of securityrelated events and this trail must be protected from unauthorized access or destruction. The Criteria is quite specific as to the events and associated information that must be recorded in the audit trail.

I do believe that Division C software controls are extremely useful in both public and private sector environments, and that many organizations are already employing them to some extent. One of the goals of the Criteria is to encourage the development of more Division C systems.

Though the DoD believes that the added trust provided by Division B and Division A are required to deter serious penetration attacks, there are many applications and environments within our organization that will be provided with an acceptable level of control with Division C security. For this reason, we would consider it a positive progressive step if more Division C systems were made commercially available off-the-shelf products.

Division B systems provide a significant increase in assurance as well as providing a tighter network of access control. I will now turn over this session to Dan Edwards who will describe features of Division B systems. Thank you.

TRUSTED COMPUTER SYSTEM CRITERIA CLASSES B1 THROUGH B3

Daniel J. Edwards

DoD Computer Security Center

The DoD Trusted Computer System Criteria was structred with two objectives in mind: 1) Allow existing systems (which typically are in the D to C1 range) to evolve into systems which support better computer security, 2) define the requirements both in terms of features and assurances which will characterize much more highly trusted systems than are commonly available today. This talk will focus on the B division of the DoD Trusted Computer System Evaluation Criteria and will show the classes grow and change as one moves up the scale from B1 to B3.

The big distinction which sets the B division apart from the C division is - Mandatory Access Controls. That is, controls on the flow and integrity of information which are based on subjects (active entities) and objects (passive entities which contain or receive information). Mandatory Access Controls call for security clearances associated with each subject and information classification labels associated with each object in the system. Once the notion of labels associated with information is introduced at level B1, the focus of the Criteria shifts to assurances that the features called for are actually present and operating as advertised. Note that all of the requirements for class C2 are either present or strengthened as we move up the Criteria rating scale in the B and A divisions.

Note that in B division systems, access to information is controlled by clearance/classification (mandatory access control) checks and by need-to-know (discretionary access controls) checks.

Class B1 - A B1 system must support both Discretionary Access Controls (Need-to-Know) and a Mandatory Access Controls (Clearance/Classification). The Discretionary access control requirements are unchanged from level C2. The Mandatory Access Control requirements with supporting labeling requirements are the single most significant new item introduced at Class B1. Labels are required for each subject and object under control of the TCB. They may be either explicit labels or implicit labels that are associated with subjects and objects based on the where they are currently located in the system. The important thing is that there be some algorithmic way to associate a label with every subject and object in the system in keeping with the overall security policy supported by the system. When unlabeled data is introduced into the system in some manner (e.g., communications line, directly attached terminal, magnetic media) a specifically authorized user must determine the correct label that the TCB will associate with that data. Furthermore, the TCB must assure that the integrity of labels associated with data is maintained. If labels associated with data could be arbitrarily changed then uncontrolled declassification of information could take place resulting in a system which could not be trusted to protect information form unauthorized access. Labels must also be associated with information that leaves or is exported from the system. Again exported labels may be explicitly attached to information exchange or implicitly associated with information based on the location or physical path used to reach the output device. In reading the Criteria, questions have been raised on the difference between a multi-level and single-level output devices. The basic difference between a multi-level and single-level device is that data must be explicitly labeled on a multi-level device while data may be implicitly labeled. That is, a magnetic tape located in a SECRET vault may be considered a single level device if 1) computer readable labels are not mixed into the data stream going to the magnetic tape, and 2) the TCB is set up to assure that no information labeled higher than SECRET can be written on that magnetic tape. It would be preferable if all data being exported from a B division or higher system were explicitly labeled. However, the Center realizes that the entire inventory of DoD computing systems will not change over night. There must be some way for output produced by B division trusted computing systems to be used on other computers. Mandating an explicit label on each logical unit of information (e.g., record, file) exported from a B division system would have significant impact on other computer systems already in place which process the data but are not currently programmed to accept or process information sensitivity labels. The Criteria also requires that the specific human readable character strings which denote the sensitivity labels be setable in some manner by the system administrator. It is obvious that the internal representation for sensitivity labels will be different from the external representation. The natural implication of the guideline on configuring mandatory access control features is that 32 bits should be set aside for the internal representation of sensitivity labels. The external representation of those labels will undoubtedly be much larger. Since we are encouraging computer vendors to make trusted computer systems widely available, it is important that each site be able to specify the human readable string that an internal sensitivity label is translated to on output.

Once labels are in place for objects controlled by the TCB, Mandatory Access Control rules can be used in conjunction with discretionary access controls to determine which subjects can access objects in the system. The Criteria requires a specific manadatory access control policy be stated and enforced by the TCB. The Bell and LaPadula security policy model is a widely accepted specific security policy model used for mandatory access control. Documentation is required to state the security policy model enforced by the TCB and the specific TCB mechanisms which enforce that policy model must be explicitly identified. The policy model need not be stated in formal mathematical terms at the B1 level, however system evaluators must be able to read and understand the B1 security policy model.

The security testing requirements for Class B1 are significantly expanded over class C2. Security testing has

two phases: functional testing where the vendors' claims about security features present are tested, and penetration testing where the evaluation team looks for flaws in the specific implementation of the TCB. It is expected that some flaws will be uncovered during penetration testing at the B1 level, however, we do not expect to find flaws which cannot be covered with relatively minor adaptations of the existing software.

Summary - The big change at level B1 is the introduction of a mandatory security policy and supporting information sensitivity labels associated with data in the system. We realize that this is a big step for the general computing public and we are seeking a computing cultural revolution by making sensitivity labels an important, user visible part of the computer system. We expect that traditional third generation operating systems can evolve to the B1 level without massive changes to the user interface.

Class B2 - The sensitivity labels for data required in B1 are extended in class B2 in that terminal users are required to be notified when the security level changes for any reason during a terminal session. This means that if a user who is cleared for SECRET information starts an unclassified process running which in turn attempts to access a SECRET file, the terminal user must be notified that he is now running a SECRET level process. In addition, the B2 level TCB must support the assignment of minimum and maximum security levels associated with each attached input/output device. The TCB then constrains the information flowing to each device to be within the security limits imposed. This feature allows an installation to force the storage of SECRET material on a specific set of I/O devices such that if operation at lower security level is required during another period, all of the higher level classified information can be more easily removed.

Level B2 Mandatory Access Control requirements are extended to all objects that are directly or indirectly accessible by subjects. This means that covert channels are addressed at level B2. Covert channels in the system are controlled by engineering to reduce the bandwidth to some acceptable point and/or auditing which will enable system security personnel to spot anomilies that may indicate attempts to signal large quantities of information along a covert channel. The notion of trusted path is introduced at level B2 which requires that the user have some way to assure that communications are flowing directly to trusted software when security sensitivity actions (e.g., logon, change process security level) are initiated. This prevents spoofing attacks where attackers leave programs mimicking the logon sequence running on unattended terminals waiting for unsuspecting users to attempt to logon while the attackers program collects the passwords and gives the user some sort of error message such as "password wrong, try again" and then vanishes leaving the user to logon again, never suspecting that the password has been collected by someone else. Note that the trusted path requirement has significant implications for terminals which use level B2 across computer networks. Even in the computer network environment, there must be some way to assure network users that security sensitive operations are handled in a trusted manner. At level B2 the Criteria requires that the roles of computer operator and system administrator be clearly distinguished. System administration clearly has security implications which should be limited to a subset of those charged with normal system operation. This separation of roles is in keeping with the principle of least privilege which says that an active entity in the system should have no more special privileges on the system than those required to carry out authorized functions. This principle is flagrantly violated in many third generation operating systems which define super privileges and then use them for relatively trivial system maintenance and operational functions. Much emphasis is placed on the overall system architecture at level B2. The basic operating system itself must be separated into protection critical and non-protection critical sections which are kept separate by effective use of the available hardware. This forces systems designers to rethink the protection critical issues in order to achieve an appropriate structuring of the operating system. The concept reference monitor which is always invoked, tamper-proof and small enough to be throughly analyzed begins to emerge at class B2. It is possible that some current third generation operating systems will evolve to the B2 level but it will take a significant amount of work to achieve the required restructuring. The B2 level requires a security policy model stated in formal mathematical terms and also requires a descriptive top level specification (DTLS) of the system to aid designers and evaluators in understanding the user interface and inner workings of the systems. The DTLS may be written in English or some other program design language. The DTLS must accurately reflect the user interface and must be maintained along with the source code as the system is modified. Configuration management of the TCB is required at the B2 level. This means that a clear trace must be maintained of who has modified what part of the system, why it was modified, and when the modifications were actually introduced into the master copy of the system. Procedures are also required to allow the generation of the system from source code rather then an evolving object version of the system. This requirement allows the evaluators to generate the system from source code and compare it with distributed object code to make sure that no unintended 'fixes' have been placed in the object deck.

Summary - Level B2 is characterized by significant commitment to the implementation of the reference monitor concept. Configuration control gives additional assurance that the system distributed by the vendor to DoD sites is the same system that the evaluators examined. Third generation operating systems may be evolvable to the B2 level with some difficulty.

Class B3 - B3 systems move an additional step closer to the technology which will characterize the very trusted systems needed for high exposure installations in the future. Class B3 features systems that demonstrate a high degree of modularity, supported by appropriate hardware. The distinction between security relevant and non-security relevant parts of the operating system is made clear by the overall structure of the operating system. The system structuring required at level B3 is so complete that few if any of the existing third generation operating systems are expected to be able to evolve to this point. The TCB more fully realizes the basic reference monitor concept in that the security relevant portions of the system are small enough to be subjected to a thorough security test and analysis. Non-security relevant portions of the operating system are

isolated or encapsulated in order to assure that any malfunction cannot damage the integrity of the security relevant portions. A formal security policy which describes the security enforced by the system is required along with a descriptive top level specification from which the the vendor provides a convincing argument that the system meets the security policy model. Level B3 is a likely target level for vendors who want to explore the edges of the verification technology used in A division systems but who do not want to fully commit to that technology yet. A carefully constructed B3 system could evolve to an A1 system or beyond. Other features called for at level B3 are an increased trusted path requirement which gives further assurance that the user (or another directly connected computer) is dealing with trusted software as opposed to being spoofed by untrusted software. Auditing requirements are strengthened to the point where security thresholds can be set on certain events to alert the system security officer that unusual activity is taking place. Level B3 also requires software or other mechanisms be provided to assure that the system can be recovered into a secure state in the (unlikely) event of a system failure. The discretionary access controls at level are also strengthened so that the discretionary access control mechanism will allow a wide range of flexibility controlled by the user along with operational convenience. The Center is currently working on a guideline which discusses the issues involved in discretionary access control to help system designers specify implementations which truly encourage people to work together in groups while maintaining the principle of individual accountability.

Summary - Class B3 systems are built according to the principles of least privilege and individual accountability. Least privilege in the sense that roles for system administrator and system security officer are defined, and least privilege in the sense that the operating system is broken up into encapsulated modules where the security relevant parts are isolated from the non-security relevant parts. Individual accountability is stressed in the increased discretionary access control and auditing requirements which allow users greater flexibility to share information subject to security constraints and work together in groups and still have sufficient audit information to hold them individually accountable for actions taken on the system. It is expected that most systems in the B3 or higher classes will be built (or rebuilt) from scratch rather than be evolved versions of existing third generation operating systems.

COVERT CHANNELS -
ALLOW COMMUNICATION BETWEEN ENTITIES IN WAYS THAT CIRCUMVENT THE SYSTEM'S SECURITY POLICY

- Covert Storage Channels: Involve direct modification of storage objects
- Covert Timing Channels: Involve modification of response time seen by user

Slide 1

IMPORTANT FEATURES OF B3 SYSTEMS

-High degree of modularity
-Isolation of non-security relevant parts of operating system
-Formal security policy required
-Increased trusted path requirement
-Audit requirements strengthened
-Recovery into a secure state
-Strengthened discretionary access controls

Slide 2

IMPORTANT FEATURES OF CLASS B2 SYSTEMS

-Implementation of reference monitor concept
-User notified of current security level
-Covert channels addressed
-Trusted path introduced
-System administrator and operator clearing distinguished
-Formal statement of security policy model
-Configuration management emphasized

Slide 3

IMPORTANT FEATURES OF B1 SYSTEMS

-Labels
-Mandatory access controls
-Expanded security testing

Slide 4

LABEL

A SEQUENCE OF BITS OR CHARACTERS THAT IS ASSOCIATED WITH A SUBJECT OR OBJECT TO DESIGNATE THE SUBJECT OR OBJECT'S SECURITY LEVEL

Slide 5

DoD TRUSTED COMPUTER SYSTEM EVALUATION SYSTEM

Division	Class	
A		Verified Protection
	A1	Verified design
B		Mandatory Protection
	B3	Security domains
	B2	Structured protection
	B1	Labeled security protection
C		Discretionary Protection
	C2	Controlled access protection
	C1	Discretionary security protection
D		Minimal Protection

Slide 6

REQUIREMENTS FOR CLASS A1 SYSTEMS AND MAJOR DIFFERENCES BETWEEN DIVISION A AND DIVISION B SYSTEMS

Dr. Carl Landwehr

Naval Research Laboratory

I am not an author of the Orange Book (I work for the Naval Research Laboratory, not for the Computer Security Center) so my view is that of a consumer of the Criteria rather than a producer. I look at it as something that I might employ sometime if I were to attempt to build a secure system. So, I've tried to aim this presentation towards people like myself who may try to use the Criteria.

The crucial fact about Division A is that A stands for Assurance. That's what the A level is all about. In my view, it requires a higher level of assurance than any of the other divisions shown on Slide 1.

As Sheila mentioned, the categories of criteria that are listed in the document are four: Security Policy, Accountability, Assurance, and Documentation (at least at the A level, there is Documentation) (see Slide 2). In fact, the A level levies no new requirements on the first two of those categories. The functional requirements for a level-A system are identical to those requirements for a level-B3 system that Dan has just described. So, you might say, "I've got a B3 system. Can I just change it into an A system by implanting a little more assurance - more tests, and a little extra documentation?" The answer is, "No." In fact, the decision to meet the level-A criteria affects the entire life cycle of a system, because those requirements, even though they do not change the functions of the system, have to do with how that system is developed.

I've tried to indicate on Slide 3 a simple view of the life cycle of a system: starting with requirements specification, then the top level or system specification, detailed specification, implementation, testing, presumably, and then operation. The notations along the sides show where level A has an affect. In the first place, as you can see, it requires more strict configuration control. Configuration control gets pushed back all the way to the design phase. You must have configuration control if you want a level-A system; you have to show that the control was in place during the design and was applied to the design documentation. I've noted a security model on the right-hand side because that's required. As Dan noted, a formal description of a security model is, in fact, required at the B2 level by the Orange Book. The main new requirement for A level is a Formal Top-Level Specification (FTLS). At the B level, only a Descriptive Top-Level Specification (DTLS) is required.

An FTLS has to be formal, and there is a definition in the back of the Orange Book of what "formal" means when applied to "Top-Level Specification." I don't think anyone has a perfect definition of that word. In any case, you have to demonstrate a correspondence between the security model and the Formal Top-Level Specification. As I read it, that correspondence does not have to be demonstrated formally; that is, you don't have to have formal mapping from the security model to the top-level specification, but the correspondence has to be demonstrated somehow. I suspect that if you can do it formally, it will be more convincing. But a lot of this has to do with whom you're convincing, and what they have in mind. The dashed lines on the side of Slide 3 represent the requirement to demonstrate the correspondence between the formal top-level specification and the detailed specifications. There is not a requirement for a formal detailed specification, and so that correspondence is probably going to be informal, unless you've done even more than is required. Similarly, there needs to be a demonstrated correspondence between the detailed specification and the implementation.

Dan talked about covert channels and described them. There is a requirement at the A level for a formal analysis of storage channels. That requirement is (the way I read the Criteria, anyway) the one place where the formal top-level specification really gets used. You can't do a formal analysis of storage channels without a formal top-level specification. So that requirement is a short sentence in the Criteria that, in many respects, actually levies the requirement for an FTLS.

Finally, in the operations and maintenance phase, there is a requirement for a trusted distribution facility. There is quite a lot there that I've gone through very quickly. You may still ask, "If B3 specifies all the necessary functions, and we have all the functions we need in B3, why bother with level A? Wonderful, you've got such-and-such documentation, but who needs it?" The reason, (Slide 4) it's needed is that we want assurance that the functions operate as intended. And the reason we want that assurance is that we want to place greater reliance on the automated controls of the system. Then we can reduce the procedural and personnel controls and operate these systems more flexibly and effectively. Without that additional assurance, we can't relax procedural or personnel controls.

Now, (Slide 5) I'll back off just a little bit, philosophically, and say, "Let's look at these criteria. Why would you want to take this particular way of gaining assurance?" The Orange Book says that we get higher assurance by having formal specifications and things like that. So I thought a little about what people do to get assurance in other systems.

The first thing people do is test. And there are requirements for testing in the Criteria as well. I don't mean to overlook those. I would categorize "test" as positive, that is, trying to demonstrate that the specified functions of the systems are there. It's the kind of thing people do all the time. There is also negative testing, where you try to see whether the system will break, or if you can break it. That seems to be one way of getting assurance: testing.

Another way is redundance. It seems to me that there are different kinds of redundance. One kind, that we've seen in the space shuttle, for example, is to say, "Here's one specification. Let's have two independent implementations of it. You do it, and you do it. And then we're going to run them and compare the answers." If the answers come out the same, even though different people implemented the specification, I may have a little higher confidence that the answer is what I wanted. At least both implementors made the same mistake. That's one way of getting some increased confidence, and I think that it is based on a kind of redundance.

Another way, the one advocated in the Orange Book is to construct different descriptions of the system and show that those descriptions are equivalent. These are, in a sense, redundant descriptions of the system. You start with a security model, which is a very high-level, abstract description of the system behavior. You must have a formal security model, even for B2, as I've already said. The A level requires an FTLS as well, and you must show that it corresponds to the security model. The correspondence may be informal, but it must be demonstrated. I think that's a kind of redundance. That's where the assurance comes from, in my view.

The guideline for testing is also strengthened for A-level systems. It says more people have to fail to penetrate an A level system than a B level system, and they have to be smarter people. You'll have to look up the details.

The next two slides (Slides 6 and 7) review the criteria that are in the Orange Book. There are additional criteria at the A level in two categories - Assurance and Documentation. Two kinds of assurance are described: operational assurance and life cycle assurance. Under operational assurance, the requirement for formal methods for covert channel analysis is levied. That is the only requirement added in operational assurance, I believe. In life cycle assurance, first comes security testing - this requires a demonstration that the implementation is consistent with the formal top-level specification. I raised the issue at one point that tests (unless they are exhaustive - a practical impossibility for large systems) can't really demonstrate that two things are consistent; they can only demonstrate that they're inconsistent. Apparently, this wasn't considered a serious discrepancy. Second under life cycle assurance comes the requirement for design specification/verification. Here, the requirement for a formal top-level specification is imposed. The formal top-level specification and the descriptive top-level specification have to include everything visible at the TCB interface. The idea here is that the trusted computing base provides certain functions. All of those functions visible to users at the TCB interface have to be called out in the formal top-level specification, and that includes functions that are not even software implemented - they could be implemented by firmware or hardware. The SCOMP FTLS, for example, includes some hardware functions.

Slide 6 also covers the third and fourth A level requirements under life cycle assurance. These requirements are more mundane; they don't push the state-of-the-art, except that they're rarely applied as extensively as the Criteria implies. The first requirement is for configurations management. There must be a configuration management system to control changes to the formal security policy model (should you wish to make any such changes), the descriptive top-level specification, the formal top-level specification, and so on, throughout the entire life cycle. The "and so on" includes design documentation. Let's back up for a second. We have a view of a wonderful system in which we have a formal top-level specification, and we have an implementation, and we have demonstrated some correspondence between them, and then we have code. We want to rely on the controls in that code. So we must be sure that the code that runs operationally is the code that we built - that it hasn't gotten subverted somewhere along this path. That is the motive for having tools to compare a new version of the trusted computing base with a previous version and show the changes. If, in turn, we are to rely on what these tools tell us, it becomes important that the tools work properly. So those tools have to be under configuration control too, because subverting them can be equivalent to subverting the mechanism for releasing new versions - it could allow a Trojan horse to be inserted unnoticed. Similarly, the material for generating the trusted computing base must be protected. The requirement for a trusted distribution facility follows from this line of reasoning. The specific requirements here are to assure the integrity of that mapping between the specification master copy and the code master copy.

Slide 7 shows the requirements added by Level A for documentation. First, test documentation is required. This must include the results of mapping the trusted computing base source to the formal top-level specification. The Criteria does not specify the form of this documentation; that will probably be determined on an adhoc basis. As part of the design documentation, the correspondence between the formal top-level specification and the implementation must be described. This description can be informal. The way that the elements of the trusted computing base correspond to the FTLS must also be documented; this too can be informal. I find this a little confusing. I'm not sure what the elements of the formal top-level specification are versus the elements of the trusted computing base. Maybe Sheila can enlighten me on that.

SHEILA: Yes.

LANDWEHR: The final requirement is for a description of the hardware, firmware, and software mechanisms strictly internal to the TCB. Elsewhere, a description of the functions (hardware, software, and firmware) available at the TCB interface is required. Here, that requirement is extended to require a description of any mechanisms internal to the TCB that are not reflected in the FTLS. I suspect the motive is to uncover sneak paths within the TCB that are not covered in the TCB interface specification. But, again, I'm uncertain exactly what the considerations were for including this requirement.

To recap, suppose I want to build an A level system. How will its life cycle differ from that of a system intended for Level C or B? (Slide 8) In my view, the formal top-level specification should be developed prior to, or at least in parallel with, the descriptive top-level specification. The descriptive specification corresponds to a conventional software design document. The formal specification should control the informal detailed specification; at the least it ought to track the changes in the detailed specification.

Otherwise, it will be difficult to show that a mapping exists between the two. For system developers this is a very important point. To get the benefits of this approach, the implementors have to understand the language in which the FTLS is written, and they have to be competent to update that specification. Otherwise, one group will write the FTLS and then another will implement it. If the implementors can't read the FTLS, they may just use the informal specifications. Differences will arise among the different specifications, some will get out of date, and demonstrating the necessary correspondences will be impossible.

Increased configuration control will also change the life cycle, as will closer controls on distribution and maintenance.

For those of you who haven't seen a formal top-level specification it's typically a collection of functions defined in a particular non-procedural notation (Slide 9). Non-procedurality is not a requirement, but that's the most common form. In any case, it's a collection of functions analogous to those you might see described in English, only presented in a more structured, less ambiguous (more formal) way. It will be a more mathematical-looking document than usual specifications. Slide 10 lists some available languages. You can read about them in an article written by Maureen Cheheyl, Morrie Gasser, George Huff, and Jon Millen, entitled "Verifying Computer Security," in *ACM Computing Surveys*, September 1981. That's a good place to begin learning about formal specifications for computer security.

Another thing I would want if I were going to build an A level system would be some examples (Slide 10). It's always easier to do something new if you have an example to follow. Unfortunately, no A level systems have been certified yet, but there are some evaluations in progress, and there is documentation available for some of those. Unfortunately, I can't tell you where to get these documents. However, I did my best in an article that appeared in *IEEE Computer* in July 1983, and I have provided some references there. You might also ask the people at the Computer Security Center, since I think that they ought to develop a library of such documents or at least provide references to them. I will point out one other recent article, by Jon Silverman, on the verification of the SCOMP kernel. It is in the Proceedings of the Ninth Symposium on Operating Systems Principles - ACM SIGOPS.

Earlier drafts of the Criteria included an A2 level, which has been deleted. I think the reason for the deletion is that people at the Center think that meeting those requirements is not within the state-of-the-art yet. What we might see in the future (Slide 11) are requirements for using verified tools to produce secure systems. We might have verification requirements on compilers, for example. We might also see some proofs, not just of formal top-level specifications, but of lower-level specifications, and proofs of correspondence between levels. Perhaps test data will be generated automatically from specifications. Dan alluded to the idea that in a more structured system one might be able to do a more intelligent job of testing. We may also see proofs of different sorts of security properties. The primary emphasis of security properties now, as was pointed out this morning, is on disclosure. There may be other properties people could formulate, and they might like to prove that they are preserved by some system.

I will close with one problem that I can't resist pointing out (Slide 12). There have been some small systems built, perhaps a thousand lines of GYPSY in length - small but, nevertheless, functional systems that have been implemented and have had their code verified, as well as their formal top-level specifications. So they've actually met the fundamental requirements for assurance that are levied by level A. In fact, they've not only met them, they've exceeded them. However, these are small, special-purpose systems. They don't provide the functions that are required of even a B1 system and they don't need to provide them. So, under the current Criteria, if I had to evaluate them, I'd probably have to rate them somewhere in level C. This, to me, is a problem. The structure of the Criteria now gradually increases what's required on all components as you advance from one level to the next. There is an increase in the formality with which the security policy is stated, in what labelling is required, in how much testing is required, and so on. There is a gradual increase in requirements in each category from C1 to C2, C2 to B1, B1 to B2, and so on until we make the jump from B3 to A. Level A primarily increases requirements in a single category: assurance. So, I see an unaesthetic difference between the way A is defined relative to the other levels and the gradual entry of the others. I'm not sure exactly how to improve this.

SHEILA: What would you see as a better rating scale?

LANDWEHR: Do you want me to propose one?

SHEILA: Yes. Since you brought it up.

LANDWEHR: I don't have a ready answer. One possibility is to have ratings apply independently to each of several axes. I think separating concerns and being able to say that a system has one level of assurance and another level of function, for example, might be useful. I think that's quite a possible scheme, though it's not the initial one. There may yet be a different color document after the orange one. (I'm speculating.)

'A' is for Assurance

A>B3>B2>B1>C2>C1>D

Slide 1

Evaluation Criteria Categories:

Security Policy
Accountability
Assurance
Documentation
Level A functional requirements =
Level B3 functional requirements

Slide 2

But...

A decision to meet Level A criteria affects the entire life-cycle of the system:

	Requirements Specification	security model
	Top Level Specification	FTLS
Strict Configuration Control	Detailed Specification	
	Implementation	
	Testing/Analysis	formal storage channel analysis
	Operations/Maintenance	trusted distribution facility

Slide 3

If B3 already specifies all necessary functions, why bother with level A?

Assurance that the functions operate as intended is crucial if we are to rely on them in place of physical and procedural controls.

Slide 4

How do we achieve higher levels of assurance in systems generally?

Testing: positive - functional
negative - penetration

Redundance: parallel developments from same specification
(e.g., space shuttle)
construct different descriptions of the system
and show they are equivalent
(e.g., security model, FTLS, detailed spec, implementation)
A Level criteria based on implementation, etc.

Benefit: completeness

Guideline for testing also strengthened for A Level.

Slide 5

What are the specific requirements added by Level A?

ASSURANCE

operational: formal methods shall be used in covert channel analysis (for storage channels)

life cycle:

security testing: show implementation consistent with FTLS

design specification
and verification: FTLS

FTLS - security model
correspondence

FTLS and DTLS to include 'visible' TCB hardware and firmware

configuration management:

a configuration management system must control changes to: formal model, DTLS, FTLS, etc. throughout entire life cycle

tools for comparing TCB tools for comparing TCB versions to configuration control

protection of material for generating TCB against modification/destruction

trusted distribution:

provide trusted facility to:
- assure integrity of mapping between specifications and code masters
- assure distributed copies match masters

Slide 6

DOCUMENTATION

test documentation: include results of mapping TCB source FTLS

design documentation: show FTLS implementation correspondence (informal)

show how elements of FTLS correspond to elements of TCB (informal)

describe hardware/firmware/software mechanisms strictly internal to TCB not described in FTLS

Slide 7

How is the life cycle different for an A Level system?

Formal Top Level Specification should be developed prior to or in parallel with Descriptive TLS

FTLS should govern detailed specification and implementation - at least it must track changes in design

Implementors should understand and be competent to update FTLS

Configuration control is applied earlier

Distribution and maintenance are more closely controlled

Slide 8

What does a Formal Top Level Specification look like?

Typically, a collection of functions defined in a particular nonprocedural notation

Languages used for FTLS's include:

SPECIAL	(SRI)
FDM	(SDC)
AFFIRM	(ISI)
GYPSY	(U. TEXAS)

(see Cheheyl et al, "Verifying Security," *ACM Computing Survey*, September 1981)

Slide 9

What examples are there to follow?

None that have been certified, but several that have successfully written FTLS's and implemented systems based on them

SCOMP - Honeywell
KSOS - FACC, Logicon
KVM - SDC
COS/NFE- Compion

(see "Best Available Technologies for Computer Security," *IEEE Computer*, July, 1983, also Silverman, "Reflections on Verification of the Security of an OS Kernel," Proc. 9th S.O.S.P., October, 1983)

Slide 10

What lies ahead?

Proofs at the implementation level

"Verified" tools (compilers, etc.)

Automated test generation

Proofs of different security properties

Slide 11

A Problem

Some small systems have been implemented and have had their code verified as well as the Formal Top Level Specifications

This is a level of assurance beyond A level as presently defined.

But these systems do not provide (and do not seem to need) all of the security functions required of a B1 system. Hence, they appear headed for a C2 evaluation.

Isn't this a problem?

Slide 12

PANEL SESSION - SECURITY REQUIREMENTS FOR COMPUTER NETWORKS AND PROFESSIONAL WORKSTATIONS

Moderator - D. Elliot Bell

D/Chief, Research and Development

DoD Computer Security Center

Panel Members:

Ray McFarland - DoD Computer Security Center

John White - The Mitre Corporation

INTRODUCTION

Three major areas of technology advance, computer networks, local area networks (LANs) and intelligent workstations are posing a significant challenge for computer security. This panel will address these three areas with an eye towards illuminating the problems themselves, the applicability of current security tools and methods to the problems, and directions for solving the problems.

Computer networking provides a method of connecting computer facilities in such a way as to allow users to make use of the resources of all the constituent computers with near ignorance of the networking mechanisms. The computer security problems that must be faced are the lack of a definition of "secure computer network"; the provision of mechanisms for supplying security labels for messages in the network; the general problem of modularizing models and systems of formal specifications; and the anomaly of protecting information in a packet-switching network while sharing the associated header.

LANs have become a very important means of connecting computer and peripheral equipment (such as terminals) with shared transmission medium. The issues are the problem of isolating information while sharing a transmission medium; and the performance issue of providing isolation (a problem not so critical in the case of normal computer networks).

Intelligent workstations offer the prospect of absorbing more and more of the data processing load at the individual workstation, freeing the central resource (or resources) from routine chores or providing the option of using a less powerful processor for the central jobs. The difficulty with respect to computer security is that a computer system configured with intelligent workstations is a conglomeration of computing equipments, each one of which must be defined, conceived, designed, and accredited as a secure computer system. The working of a set of interrelated secure computer systems has not been addressed satisfactorily in a conceptual sense and the general difficulty of modularizing models and formal specifications arises here also.

Across all three areas, the advances of computer technology in computer networks, local area networks, and intelligent workstations call for conceptual advances in modeling, specification, and formal verification to handle the relatively more complicated computer systems at issue. Adding security features after the fact has never been successful, initial work on the fundamentals of these areas (protocols and so forth) must be undertaken to forestall painting ourselves into a corner technologically.

Ray I. McFarland, Jr.

Local Area Network (LAN) DEFINITION

Cable broadcast technology results in the ability to communicate reliably over short (local) distances at high speeds.

Characterization of cable communication types:

1. Multidrop line
   - link level protocols only
   - similar to host computer terminal lines
2. Closely Coupled systems
   - sharing of operating system tables and routines
   - characterized by "network operating system" or "distributed processing system"
   - protocols closely associated with Operating system like inter-procedure calls (??)
3. Loosely coupled systems
   - operating systems not shared, are autonomous in terms of control and resources
   - protocols similar to ARPA pioneered protocols (TCP and IP)

LAN PHYSICAL ENVIRONMENT

Assumptions for purpose of this talk

- network is physically protected from external attack
- when exposure exists, full period TFS is sufficient

- I am addressing BIUs/network interfaces, not hosts attached thereto except where explicitly noted (in last part of talk dealing with multiple models and verifications forming an MLS system)

LANs vs OPERATING SYSTEMS

motivation: formal security models and techniques developed in communities today are primarily based on characteristics of operating systems

operating system characteristics

- dynamic creation and deletion of processes
- dynamic interprocess communication paths
- data sharing between processes, subject to security constraints - normally represented by the DoD classification system
- operating system controls
  - tend to be processes initiated at security level of data being processed
  - access control is centralized
  - denial of service solutions NOT well understood

LAN characteristics

- able to statically define those processes required to process a message stream
- able to statically define processes which are required to communicate with one another
- prevention of data sharing, i.e. ensuring data integrity of a user's connection, regardless of security level
  - user data not corrupted by processing
  - user data not mixed with other's
- network control information/processing
  - exposure to users
  - protected from users
- access control is distributed
- solutions to denial of service are "better understood," "easier - but NOT easy"
- some network security requirements may be protocol oriented
  - example: flow control for denial of service

Modeling issues - what' s missing:

- ability to incorporate security functions that are NOT just based on security level of data
- NOTE: One normally associates formal modeling and techniques and trusted computing bases with requirements for multi-level security. Single level systems which implement security functions in computers also need to be trusted, which implies formally addressed as well.
  - formal modeling today primarily deals with security label oriented segregation
  - today, we CAN use segregation techniques to control/protect security functionality, even if we can't formally verify the functionality itself.
- must recognize that gross labeling approaches (i.e. limited to DoD classifications) may be inadequate for networking.
- access control algorithms for user access to network resources (network access controller concept)
- denial of service

Modeling issues - what' s needed:

- models need to be defined for network security which address security functionality beyond segregation by security label (e.g. network level access control decisions)
- different or more restrictive approach to data sharing based on finer grained security labels for data integrity of user connection (or other viable approaches for providing data integrity)
- where protocol functionality plays a role, we need to define and integrate into the formal model and techniques aspects related to protocols (e.g. temporal logic work oriented at protocol functionality)
- where multiple MLS systems are connected together by an MLS LAN we need to allow ways for verifying system level security from the components formal models (NOTE THE PLURAL) and specifications
- flow control modeling (or others) for denial of service

What we can do today:

Unique Label per user connection concept definition

- each new user connection gets unreusable unique security label.
- processes are statically defined (no dynamic creation/deletion of processes)
- process IPC paths are statically defined
- processes are "memoryless" from the processing of one "packet" in a message stream to the next
- processes can only process one unique label at a time
- "data containers" can only hold "packets at one unique label at a time," and are also "memoryless," or provide sufficient segregation of data with different labels

Interpretation

- we can rely on the security label orientation of formal models today

- at any one time, a process is at the unique security level of the packet being processed

- over time - the process appears to be multi-level, but is unable to cause a security problem by the "memoryless" requirement

- data cannot be intermixed between user connections

- processes can't do it by above

- containers can't do it - hold only one type at a time or isolation techniques exist

- DOES NOT eliminate the need to check the standard DoD security labels at interfaces between MLS systems (and system high systems attached thereto)

Not modelable security functions (e.g. flow control)

- can still be included in TCB

- should be isolated from interference by other processes using normal process segragation techniques

John White

As Dave said, I'd to talk briefly about what the use of intelligent workstations does to how you implement security in a system - what kind of impact that has. I have a mental picture derived from what we think of as a classic security architecture, that's more or less centralized. We have heard a lot today about evaluation criteria which don't necessarily apply specifically to that kind of system, but I have a mindset that puts them on a system with a centralized computing and memory power, and a bunch of terminals and other devices hanging out on the side. We've got the problem now that this computing power, this intelligence, is cheap; you can spread it around, and everybody wants one on his desk (for good reasons or bad). So how does that affect your whole approach to security?

First we need to consider, what do you do with this device? You've got a computer in your hands now. Obviously, you can do things like text processing locally. You may want to do spread sheets. People have discovered wonderful things that you can do with spread sheets that were somehow very difficult to do before. You bring in some data and massage it locally, and produce a result. Maybe you want to do geometric manipulations of some modest amount of data, another thing that you can do locally. The point with all these activities, as you might expect, is that it's the data that's the problem, not the processing. You'd like to be able to use commercial software that you can buy off the shelf to do the processing. Just as in centralized systems, it's the data that really needs the protection. So that doesn't seem to be particularly different.

It just means now that this device that's sitting in the user's office - his chosen mode of operation - has got to have some substantial blast of data in to do some kind of manipulations on that data for a while, and then have a fairly substantial blast of data back out. That's an image of how the operation works, and that's going to have some implications. One of the first things that happens now, we recognize, you've lost some physical control over the hardware. In a centralized system, you have a computer center. You recognize the people who are specifically dealing with the hardware in the computer center - you need to have trust in them. Now, all of a sudden, we've got hardware all over the place, out in people's offices. What does that mean? That changes some of the rules a little. In particular, multi-level or compartmented operation, within the workstation itself, doesn't have much meaning anymore. If the data's in a guy's workstation, he's got control of it, and you can't do anything about that. So speaking just of the box itself, it can only operate system-high. But it means that the requirements to meet the system-high mode of operation become more difficult.

I've felt in the past that system-high involved a certain amount of hand-waving. I'm not sure where in the evaluation criteria you might say system-high operation would have to be. But you say, well, I've got labels on things, and the people who are dealing with this material have sufficient clearance to see it anyway. So we really don't have a problem. Now you've got this workstation that's got several levels of data in it. It's got a very high bandwidth pipe connecting it to the rest of the system. You need to be very confident that it's not streaming in data at one classification level, and streaming it right back out at a different classification level. That's the kind of thing that a multi-level system can't tolerate. It means you need some real controls in the system. The classic system-high operation focused mainly on labeling and auditing, because you had physical and procedural controls. Now you've got a very powerful, high bandwidth electronic connection. You've lost some of those controls, and you need to treat the implications of system-high more seriously.

Another problem area is local memory. This workstation has some megabytes or tens of megabytes of storage, whereas a terminal might have had enough to refresh a display screen. What do you do with that stuff? You can't leave it lying around on the system overnight; at least, you don't like to. You've got the cleaning lady coming around at 3 a.m. and dumping your memory. Even if she's been cleared, you just don't like to expose yourself to that kind of thing. On the other hand, you don't much like removable media either. You might have people tearing around with a few megabytes of classified data in their pockets. And finally, you might just clean it out, flushing it back to a central system. Clearing out memory is unattractive from the point of view of the time and/or bandwidth it takes. I certainly don't have an answer for the local memory problem. None of the ones that I know is terribly attractive.

User interfaces; these devices do have power to operate in stand alone mode we assume. You can envision a workstation connecting and disconnecting logically from the system. You'd like its security controls to remain, to gracefully cross that transition. A user should identify himself once to the system, and from then on he shouldn't be constantly logging in to other components of the system as he connects and disconnects to them. That puts some requirements on him. Switching between external interfaces, you've got a nice high-bandwidth path for gross leakage that you've got to be concerned about.

Another concern, of course, is the issue of local hard-copy printout. That's something that gives security auditors great heartburn. Conceptually, you can do the same thing with a screen that you can do with a local printer. But somehow the ability to tuck it into a pocket and lose it, or walk off with it and toss it in the trash can is something that you'd like to avoid, or at least have some control over.

I think that, with all those problems, the advent of intelligent workstations also offers some very appealing opportunities, in that we're still dealing with relatively simple and primitive operating systems: compare MVS to MS-DOS. They're a couple of orders of magnitude of complexity apart. You have the opportunity to build systems, multi-level or compartmented systems, taking advantage of putting the security controls in the intelligent workstation. If you really need to share data, why not put the controls at the point where it's shared, at the workstation? If you had different data that you really wanted to keep separated except under careful control, you could put it in physically separate data base systems. Then, when you've connected those to your intelligent workstation, let the smart security decisions be made there where the system is smaller, more controllable, more conceptually tractable.

I think that it may well be that the technology of distributed systems with workstation power could let us get into some genuinely multi-level or compartmented systems that we may not reach with the large-scale mainframes. I think that they give you the potential to have more robustness of security controls. Too often you find yourself relying on a single point of control in a secure system; and if there is some imperfection or failure in that single point, then you essentially open the floodgates. If you're smart enough, and can distribute the security controls around the system in away that makes them complement each other, rather than just doing the same job multiple times without adding any value, we could get systems that could be much more substantially trusted than any we have today.

KEYNOTE SPEAKER Day 2

TRANSFERRING COMPUTER SECURITY TECHNOLOGY FROM LABORATORY TO MARKETPLACE

Barry Schrager

President

SKK, Inc.

President and founder of SKK, Inc., Barry is also the originator, designer, and developer of the ACF2 security software system. More than a decade of work in the data security field convinced Barry that the available security products did not adequately address the growing requirements of the computer industry. So, in 1978, Barry and his colleagues, Scott Krueger and Eberhard Klemens, formed SKK, providing a vehicle for the development of an innovative security software system-ACF2.

Prior to forming SKK, Barry served as Assistant Director of the Computer Center at the University of Illinois at Chicago Circle, where he was instrumental in the development of their computer security program from 1969 to 1978. He was also manager of the SHARE Security Project from 1969 to 1975, when a series of SHARE requirements for data security were presented to IBM. He was manager of the SHARE MVS Group from 1975 to 1977, was deputy manager of the SHARE Basic System Division from 1977 to 1978, and was appointed to the SHARE Advisory Council.

Barry received a NATO grant in 1971 to participate in the Institute on Micro-Programming at the University of Grenoble, France, and in 1966 worked to develop automatic chromosome analysis by computer - a joint project of the Argonne National Laboratory and Presbyterian St. Luke' s Hospital.

Barry has completed doctoral research in computer security at Northwestern University where he received a Master' s Degree in Applied Mathematics, and holds a Bachelor of Science degree in Physics from the University of Illinois at Chicago Circle.

When I was asked to give a presentation today by Dr. Dennis Branstad, he said that since he had heard that the data security package ACF2 was developed at a university, I would be a knowledgeable speaker on today's topic, which is "Transferring Computer Security Technology From the Laboratory to Marketplace." Interestingly enough, I think he was right, but for the wrong reasons, and I will explain this later.

But first, let me introduce myself. I am Barry Schrager, President of SKK, Inc., the company that develops, maintains, and supports the software data security products, ACF2/MVS, ACF2/VS1 and the recently introduced ACF2/VM. I was the original designer and one of the original authors of ACF2 -- The Access Control Facility which was introduced for MVS systems in early 1978. There are now almost 1100 installed sites worldwide using one of the ACF2 products, many of which are installed within our Federal Government and Department of Defense. Federal Government sites include the Office of the President, the House of Representatives, the Library of Congress, the Departments of Agriculture, Energy, State, Transportation and Treasury, the entire Postal System and all of the Federal Reserve Banks. Sites also exist within the Army, Navy, Air Force and other DoD locations. SKK currently has about 100 employees located in four offices, Chicago, London, Munich and Sydney. About half a dozen agent organizations represent us in other parts of the world.

In the news lately, because of the congressional hearings on computer crime, have been two individuals, Neal Patrick of Milwaukee and Susan Headley of Los Angeles. Both were part of teenage groups that gained unauthorized access to computer systems located throughout the country and they both agreed to cooperate with the authorities in return for immunity from prosecution. The Milwaukee group made the headlines because, among other things, they disrupted the processing of information on the Sloan Kettering Cancer Institute's Digital Equipment computer system and also accessed information at the Los Alamos Nuclear Testing Facility. The Milwaukee group's activities and intent, although unlawful and unethical, were relatively harmless. But they did bring to our attention examples of computer systems in this country where what I would classify as gross negligence was taking place. The operators of these systems did not even bother to alter the standard identifier and password which are supplied by the equipment manufacturer and shipped with the system.

However, the Los Angeles group should give us all some cause to reflect on the youth of this country and on the vulnerability of our computer systems. Their goal was to shut down the telephone system for the Los Angeles metropolitan area. Now this is big-league computer terrorism.

Closer to this community of computer users are stories similar to one that appeared earlier this month in the Chicago Tribune entitled "Defense computer system

cracked; collegian held." Ronald Austin of Santa Monica, California was arrested and charged with fourteen felony counts of maliciously accessing a computer system. Using a local telephone connection, Austin gained access not only to local computer accounts, but also, through the UCLA system, to the ARPA network. The complaint said Austin gained access to more than two hundred computer accounts at fourteen locations. Some of the installations to which Austin allegedly gained access are the Naval Ocean Systems Center in San Diego, the Naval Research Laboratory in Washington, the Norwegian Telecommunication Administration in Norway, SRI International in Menlo Park, and the Rand Corporation in Santa Monica.

Another item in front of the public today is the television series "The Whiz Kids" on CBS. In this series, a handful of youths proceed to gain unauthorized access to major computer systems, supposedly in the name of the public good. Although the series is far-fetched, it still does romanticize the idea of computer tampering and could serve as a model to some youths who wish to exploit their computer expertise in a manner which is negative to our society.

Suddenly the eyes of the general public are on computer crime and we in the data processing community are being asked some serious questions about the vulnerability of our data processing systems. I remember in 1971, when I was appointed Manager of the SHARE Data Security Project, that the only people interested in data were Department of Defense installations and some universities. Why universities? Because, as usual, universities were on the forefront of the industry. Home computers were relatively unheard of and were only spoken about by the futurists of computer technology. Companies with major interactive timesharing systems or dial-up access were rare. But universities were introducing interactive timesharing services. They had large numbers of intelligent youth who were very limited in the amount of computer resources universities could afford to give them legitimately, but were relatively unlimited in the amount of time they had to address the problem. And only very primitive security controls existed. For the university computer service bureau, security was a matter of survival. Universities had to be able to provide stable and reliable services which included prevention of outages and protection of data and resource usage.

In case you had not guessed, my background is with a university, the University of Illinois at Chicago Circle, where I spent almost ten years on the staff of the Computer Center. In 1969 we introduced interactive timesharing, via the Conversational Programming System, CPS. In 1971 we introduced the IBM Time Sharing Option, TSO as it is more commonly referred to. We had our equivalents to Neal Patrick and Susan Headley back then. One organization was called the Crack Computer Club, whose goal in life was to crack the security of the computer system. For the most part, there was no malicious intent by any of these kids. They wanted additional computer time; they were intrigued and wanted to learn much more about the internal workings of the system. They bought their own manuals from IBM; they wrote their own dissassemblers and other utilities for determining the flow of control in the operating system and our security extensions; and they had a very good network for transmitting information between themselves. But there were several incidents where some of them were malicious and destroyed data that the system needed to operate with. The University then was just a microcosm of the situation that exists today on a nationwide level. Instead of a small group located in one place, the nationwide networks such as Tymnet and Telenet along with electronic bulletin boards provide similar interaction and exchange of information.

So when Dr. Branstad mentioned that I would be a good speaker for today's theme of bringing technology from the laboratory to the marketplace because the roots of ACF2 started at a university, it's just not quite the case. I just fought these battles over then years ago, have the scars to prove it, and early on developed certain countermeasures to protect my installation's data processing service from unwanted disruption.

Unfortunately, the countermeasures we used inhibited the usability of the system by providing very restrictive conditions under which data could be shared. For example, it could be completely private, or shared within the specific project, or completely public. The development of the algorithmic process now used by ACF2 to control the sharing of data was developed as an academic exercise as part of my doctoral dissertation research for Northwestern University. It was then juryrigged into the University of Illinois system and operated successfully there. The software was then redesigned and rewritten to commercial standards and combined with resource management and other controls and ACF2 as a product was developed. This was done with the technical and financial support of the London Life Insurance Company located in London, Ontario, Canada, early in 1978.

It is in the algorithmic pattern matching process that the product truly was developed in a "laboratory" environment away from commercial pressures. I knew the specifications that I wanted the final product to have. This included not only algorithmic grouping of individuals and data, but it called for the current operating environment's being a set of control parameters which determined whether access should be allowed or denied.

Many of the requirements that were part of the original specifications for ACF2 were the result of input from many helpful people. In 1973, when I was Manager of the SHARE Security Project, we produced a series of requirements for submission to IBM which called for Centralized Resource Control, a Common Uniform Identifier to be used for all types of access to the system, algorithmic grouping of both users and data, and support for Designated Interface Programs which were to be the controlled paths between users and data. Some of these requirements had even earlier roots in the original design of the MULTICS system. The main contributors to the SHARE requirements were Lew Bethards, an auditor for the Federal Reserve System with experience in commercial data processing going back into the early 195Øs, Maria Davis of Calspan Corporation, Bill Murray, a leading security expert from IBM and one of the security designers for IBM's internal Advanced Administrative System, and myself.

In 1976 IBM announced the availability of the Resource Access Control Facility, or RACF. Unfortunately, RACF did not meet the requirements proposed by the SHARE Security Project in 1973. IBM was very defensive

over their introduction of what everyone will now admit was an inferior product at that time.

I took it as a personal challenge to design a system that would meet the SHARE requirements. Roughly five months of part time work on my own time were spent in the design of the external specification language, the internal textual representation, the prototype compiler and interpreter. I doubt that many commercial ventures would have allowed for such a great investment, the more straightforward, but less flexible, solution would have prevailed, such as in the list approach used in IBM's RACF.

In my opinion, the success of ACF2 was based on the fact that it was not conceived in a laboratory. It was developed to meet the stringent needs of a real community of users that required high security, a great deal of flexibility in the sharing of data, and a very high quotient of "user friendliness" as it is now called. It also had as input the research I had done in data security systems and the real world experience that people like Maria Davis, Lew Bethards and Bill Murray could bring to it. After the prototype was operational and London Life Insurance insisted that we develop it into a commercial product, we received a great deal of invaluable guidance from Ron Murray of London Life and Bill Griffin and Bill Hartman of General Telephone and Electronics Research Laboratories.

I do not want to give you the impression that ACF2 was developed simply as a reaction to specified need. ACF2 was a highly advanced technical development that assimilated the needs proposed to us and used the expertise of the authors to produce a technologically advanced product. ACF2's advanced function, flexibility, ease of implementation, and low cost of operation in terms of both people time and machine resources established the market for commercial data security products. It was only after ACF2 proved that many companies would purchase data security products that IBM started investing more money into improving RACF. It was only after both ACF2 and RACF established that there was a large market for data security systems that CGA invested money to develop Top Secret. If nothing else, this is a perfect example of the free enterprise system and shows how free competition improves the participating products.

The theme of today's session is "Transferring Computer Security Technology from Laboratory to Marketplace." There has to be a need in the marketplace for this technology. In the case of ACF2, we knew there was a small need for universities and service bureaux. It turned out that the climate of the late 197Øs and early 198Øs made the need for a system like ACF2 much more universal. When we started SKK, we hoped to sell ACF2 to about forty installations per year. In fact, we have averaged over two hundred per year, and this year we should add about three hundred installations to the ACF2 user community. But if no need can be foreseen, or even rationalized, do not even bother trying to transfer the technology to the marketplace. No one will buy it.

This is especially true for computer security technology. Commercial installations look at computer security software and devices as an answer to some risk they perceive exists. They are not about to spend money on computer security technology unless they perceive they have some exposure to protect against. They would rather spend money on productivity aids, data base systems, and a multitude of other things that will more visibly improve their services or reduce their costs. They also will not look at computer security technology that will negatively impact the usability of the system more than a little bit. They will not look at computer security technology if it requires the forfeiture of their existing investment in computer software, data, or procedures.

I would like to focus for a moment on the conference theme which is "Trends in Computer Security" so I can cover some topics which we have talked about at SKK for some time now. One of the major problems we see in the computer security area is that there is very little pressure being applied to the vendors of software other than to those directly involved in security products. For example, how many installations require that a security audit of all software be done prior to any purchase? It does little good for our customers to come to us requesting a security interface with an application or system product if the other product does not provide a security interface facility or exit point. Good security takes a cooperative effort of all the software vendors, and sometimes, it takes a great deal of persuasion of the non-security product vendor in order to get resources committed to incorporating the facility and eventually produce an interface.

One example is the ACF2/Panvalet and ACF2/PanExec interfaces. Panvalet and PanExec are software products of Pansophic Systems, located in Oak Brook, Illinois, just about twenty minutes from SKK's Rosemont, Illinois location. We first met in 1979 on the specifications we needed in the Pansophic products in order to jointly provide proper security for installations with both vendors' products. In 1983 Pansophic delivered systems incorporating these specifications. Believe me, if more companies would have refused to purchase their products without these security interfaces, it would have been done in a lot less than four years. However, to our knowledge, Panvalet's major competitor, ADR's Librarian, does not have any security exits at all and they have not even begun an implementation of them. Another example is the ACF2/IDMS interface. IDMS is a product of Cullinet. We first talked to them in 1981, and the interface was available in 1983. A little better, only two years.

Security has to be the concern of the whole data processing industry, not just the data security software vendors and portions of the hardware vendors. It is you, the customers of data processing products that have the most influence. If you make it clear that you demand security and security interfaces in all the products you purchase, then you will have adequate security. If you do not demand this, there will be less than adequate security in parts of your systems.

The future of security in the data processing industry rests with all of us. We cannot be complacent and let the Neal Patricks and his Milwaukee friends gain access to computer systems because of negligence. There must be a security awareness in our data processing installations. We cannot be complacent and accept software that does not meet security standards, for it leaves us open to the Susan Headleys of this world who are intent on committing

computer terrorism and to the internal employees who would take advantage of less than adequate security mechanisms for their own personal gain.

Since this conference is being held at the National Bureau of Standards, security standards are a topic that should be mentioned at least briefly. First of all, no product should introduce any integrity exposure to the operating system or to other products on the system. Some products are still being shipped with general operating system interfaces that allow the caller to gain control in an authorized state. These devices were in high use ten years ago, but by now vendors should have redesigned their products to use standard operating system interfaces and features that allow the needed function to be performed without creating some liability.

In addition, products that perform data base functions for storage and retrieval of data or even programs should maintain ownership information on each data structure or program. This is very important for systems that allow program storage or data structures to be dynamically created and destroyed.

Products that support multiple different users concurrently should provide supporting control block structures which will allow a security system to differentiate which functions are being performed by which user. Operating system security products cannot look within a complex product that supports multiple concurrent users and differentiate which processes are being done on behalf of which user. In addition, there are internal resources of the product itself that should also be controlled that the security system would never be able to see.

For all these reasons, many of the products in use by data processing installations today require security interfaces. By security interfaces I do not mean only ACF2 interfaces or RACF interfaces. I mean defined subroutine exit points where all the information about a specific operation has been collected and a decision can be made as to whether the process should proceed or not. This information would include the user identification, the requested resource name, the resource owner, if that is pertinent and known, and so on. With such exit points, it is relatively easy for an interface to be written to any security product. This consistency would allow for a good level of security. For some installations represented here, it is also possible that the security provided for a specific function by a security product may not be sufficient and you may wish to extend this security by additional verification procedures, etc.

These are the reasons that industry security standards are so important. I am not suggesting that specific calling sequences or even information be specified in the standard. But it is important that the "gestalt" of what is needed should be communicated so that vendors can apply the ideas to their own products and thereby develop a set of security exit points, control block structures, and information that will allow an installation's security standards to be implemented.

This may seem like a great deal of work, but there is an interesting benefit to all this. Implementing computer security reduces exposure both to insiders, who will manipulate the system for their own personal gain, and to well-intentioned users, who accidently could cause damage. Implementation of computer security forces an installation to reevaluate its policies and procedures, enforces the new policy, and thereby prevents much of the accidental destruction or corruption of data.

Thank you for your interest this morning. I wish you a successful conference.

BEYOND A1: THE R&D CHALLENGE

George Jelen

DoD Computer Security Center

You have just heard a description of where the Computer Security R&D Program is and where it is going in the next couple of years. One can view this program as pursuing five areas of principal thrust. (Slide 1)

The first area focuses on operating systems security, aimed at determining how to design and build a trusted computer. The work in capabilities-based architectures is an example of this thrust area.

The second thrust area is in data management security which picks up on the summer study of 1982.

Next there is network security. We know that both communications security and computer security are necessary for network security, but what we do not know is to what extent they are sufficient. There seems to be a need for something over and above each of them - some elusive "value added."

Another thrust area is hardware and software integrity. Suppose that we did get to the place that we could determine that a computing system was secure. How could we ensure that it stayed that way? And how do we ensure that the system that we intend is the one that actually gets built? The problem here is not one of proving that the code (or the hardware) does all that it is supposed to do. It is the far more difficult problem of proving that it does not do anything that it is not supposed to do.

And finally, we are working in the area of formal models and with automated tools required to establish the necessary correspondence between model and code.

Considering the current number of qualified researchers inside and outside of the government, - people available to do the work - these thrusts represent an ambitious program. Yet, considering the threat, it is woefully inadequate.

This country has recently been awakened to a computer security problem by the well publicized activities of the "Milwaukee 414" group. What they were able to do certainly represents a serious threat all by itself, but it is not representative of the threat *we* in the DoD face. Rather than that of a bunch of thrill-seeking high school students, the threat facing the DoD and the Intelligence Community is that of a well-financed foreign intelligence operation, and this foreign intelligence operation is not going to content itself with simply trying to access U.S. computers via dial-up phone lines! In fact, most of us in the business have come to believe that the more serious threat is not penetration at all, but subversion - both of the hardware and of the software. The difference is profound. Penetration tries to uncover and then take advantage of inadvertant latent flaws in the system. Subversion, on the other hand, involves the deliberate planting of flaws. Unfortunately, there does not seem to be a widespread understanding or acceptance of the subversion threat, and there is virtually nothing in the near-term R&D program which sets out to address it.

(At this point, I should probably point out that I realize that computers are not the only equipment subject to subversion. It is just that my talk is about computer security R&D, so my focus is on threats to computers.)

At the moment, and probably for the forseeable future (at least the next couple of decades), we seem to know a great deal more about how to render computers insecure than we do about how to protect them. Except as a way of helping to develop better defensive measures, I do not expect to see a great deal of computer security R&D going into penetration methods. Rather, I see our objective as trying to narrow the widening gap between offense and defense and I have therefore established this as our first, broad, long-range goal. The achievement of this goal is likely to depend upon the availability of adequately secure products - products for all of our applications which are adequate to the threat that each application faces, and products which offer a variety of functionality. What we need, then, is an array of products. (Slide 2) We need general purpose computers, from micros to super computers; we need special purpose machines -- to serve as intelligent work stations, communications processors, message systems, etc.; and we need a diversity of specialized secure software such as secure data base management systems, for example. And we need these products to be so trusted that they can offer protection against hardware subversion. In particular, we need products beyond the "A1" level. This, then, is the R&D challenge - to create these products.

The basic building block for trusted computers has been the security kernel. There have even been a few trusted systems built around this technology, but the amount of trust that we have felt confident in placing in these systems has been limited. As most of you know, our degree of trust depends upon how close we can come to establishing a correspondence between a formal security model, expressed in some formal system, with the actual machine code. The closer we can get to the microcode level of the machine, the higher our degree of trust. There have been a few isolated cases in which this kind of correspondence has been established down to the level of a high-order language - but our experience base even to this level is very limited and there is virtually no experience beyond the high order language level. Although we desperately need this experience, we will probably have to go after it one step at a time. (Slide 3)

The first step after Class "A1" is to ensure the availability of sufficient verification tools to prove that the top level specification of the trusted computing base is consistent with the security model. Presently, Class "A1" only requires that this consistency be formally verified when possible - in other words, when verification tools exist - and permits informal methods otherwise. The first step, therefore, is to close this loophole by ensuring the ready availability of appropriate tools. This has been a long-standing goal of our R&D program and it continues to be.

The next step is to verify the implementation to the level of the higher order language. (To get to this step may require an intermediate step of verifying a set of lower level specification statements.) As I mentioned earlier, the verification to the higher order language level has been done in a couple of isolated cases as a research exercise, but it is far from a routine procedure and there are not products yet available for which this verification has occurred. After this step, we must verify the assembly level code. We can reach this step either by verifying the particular compilation or by verifying the compiler that performed it. Although the first might be easier, verifying the compiler might have greater long-term benefits. Next, the verification must be extended to the microcode or machine language level. Again, one could verify either the result of the assembly or the assembler itself. This entire process is one of confidence building. I wish to be able to trust this system I am building, so I devise these formal methods to convince myself that such trust would not be misplaced. One problem with it, however, is that it begs a most important question, "How do we verify the verifier?" How do we prove the correctness of the verification tools themselves? It does not seem quite cricket to use a tool to verify itself. So, we must devise some other method of gaining the requisite confidence in our verification tools. Assuming we are able to do this - and I have not yet heard anyone suggest how - we should have reached the point that all software ought to be as secure as the model we began with, which, I might point out, could have been tampered with itself. Some knowledgeable person could have added some undesirable functionality to the model, although one would hope that this would be more easily noticed. At any rate, there still remains the hardware problem. We would still need reliable means of either preventing or detecting subversion of the hardware. Again, the problem is not one of assuring sufficient functionality; it is the more difficult problem of detecting extra functionality.

At this point, I have a way of displaying the R&D challenge, graphically. (Slide 4) The display takes the form of a two-dimensional matrix or table. Along the left side, I list all of the steps in this hierarchy of increased confidence. Along the top, I list all of the functional capabilities I want. The elements or boxes in the matrix, then, present some graphical indication of the extent of the R&D challenge. As you notice, there are a lot of boxes!

So, how do we fill all these boxes? How does the government see to it that all the products represented by these boxes get built? The government will surely require the efforts of industry, but the question I am asking relates to the government's strategy. In the way it relates with industry, the government has a number of strategy options. For example, in COMSEC, the government has consciously decided to fund developments itself and to classify such developments. In computer security, on the other hand, an assumption has been made that the government could not afford to fund the necessary developments, and has instead been pursuing a strategy - which we have called our "Computer Security Initiative" - of trying to coax industry into creating secure products on their own, in an open, unclassified environment. Having pursued this strategy for several years, it is now clear that although an improved set of products is likely to emerge eventually, these products are going to be slow in coming and are likely to fall short of what we really require for some applications when they do come. More specifically, the present strategy will do well to produce a full line of products at the "A1" level. The problem, as I have said before, is that many (albeit a small percentage) of our current applications require protection far beyond this. In particular, some of today's applications require protection from both hardware and software subversion, and the mechanisms offered at Level "A1" do little to thwart the subversion threat.

My remarks should not be interpreted as recommending an abandonment of the strategy embodied in the computer security initiative. Most of the applications within the government and even within the DoD do not require products beyond the A1 level and probably would not for several years. I believe that our current course can produce A1 and below products. The government has no interest in competing with industry to produce these products. What I am saying, however, is that we now believe that some of our more sensitive applications require protection beyond that which class A1 offers - protection against subversion - and if the government is to acquire some of those higher level products, it will probably be necessary to take more direct action to get them.

As I have said, one of the major factors separating Class "A1" systems from those in the higher classes is a formal verification of the implementation. If we could formally verify that the machine level code of the computer's operating system and of all other critical software was in one-to-one correspondence to some formal security model, we would have gone a long way toward offering a level of protection at least against *software* subversion. If our model were sufficient to provide security, then so should our machine level code be. although this code-level proof would still not do much to protect against *hardware* subversion, it would be a significant achievement, nevertheless. The problem, of course, is that such code level proofs are far beyond the capability of the crude verification tools that are available today. Not suprisingly, industry has not shown much interest in entering into what would amount to a research effort to produce a main-line product, particularly when it perceives a fairly small market for such a product. We therefore need to somehow detach ourselves from this total dependence on industry, and it is for this reason that we now find ourselves reevaluating our single-thread strategy. With respect to who funds the development and with respect to whether that development should be classified or unclassified, we seem to have four options. (Slide 5)

The present COMPUSEC strategy, the "Computer Security Initiative," is here called Strategy 1. Strategy 4 is the traditional COMSEC strategy. Note that there are two other choices. Strategy 2 combines industry risk-taking with classified development, and Strategy 3 combines government sponsorship or risk taking with unclassified development. Significantly, one can find successful examples of each of these strategies - occasions when the government has selected one of these other strategies *and* they have worked. We probably ought to be exploring their application to the computer security problem.

A significant point here is that there is no particular reason that we have to limit ourselves to but one strategy. We are now looking at all of these alternative strategies and

we would expect that, over the next few years, we will move toward some kind of mixed overall strategy - attempting to accrue the advantages of different individual strategies. Since this is likely to have some profound effect on how we relate to your companies and organizations in the years ahead, we would like to have your input.

I said earlier that my first, broad, long-range goal was to narrow the widening gap between offense and defense. There are two others. (Slide 6)

It is obvious that communications security and computer security are interrelated and interdependent. In fact, they are recursive. We need to understand these interdependencies much better than we do now or we risk failure in achieving either one of them. Specifically, we need to better understand what we usually refer to as "network security" and we need more research and relevant development in how to achieve it. Although I am talking here from a computer security point of view, we clearly need relevant R&D to come from a COMSEC perspective as well.

My final long-range goal - after we are well on our way to achieving the first two - is to reduce the cost of the mechanisms we invent to secure computers. At the moment, any measures capable of protecting against subversion would appear to be extraordinarily expensive. My own opinion is that the life cycle cost of a computer which could stand up to hardware subversion is likely to be two orders of magnitude higher than that of an insecure computer of like functionality. Few in the government are prepared for this kind of cost differential. In fact, many would argue that computer security is simply not worth such a price. I do not happen to feel that way myself but I certainly admit that, once we know what to do, we need to discover ways of achieving the same result at more reasonable cost. Hence, this third long-range goal. It must be in third place, however, because if we allow cost to be much of a consideration as we tackle the first two goals, I fear that we will not achieve either of them during our lifetimes.

There is, after all, an alternative. The alternative to spending the necessary money for an aggressive R&D program is to start pulling some plugs and cutting some wires. We could go back to dedicated or nearly-dedicated computers. We could disconnect some of the large networks of computers with wide ranges of clearances and compartments, and return to a simpler world that today's security technology could probably cope with. But I sense - and I suspect so do all of you - that no one wants or is even willing to go back. Everyone wants the added functionality which today's networking technology offers. At the same time, most want to be able to operate securely. To achieve both, we will have to be willing to pay the higher price that adequate security will require or we simply will not have it.

The problem facing computer security R&D is not one of further refinement. Rather, it seems to me, it is one which requires some bold, creative initiatives. I am most bothered by the extent to which we in the computer security community have comforted ourselves with limited expectations. Surely, if one sets his goal low enough, he can probably hit it. Unfortunately, we in the COMPUSEC business do not have the luxury of setting our own goal. Our adversaries have done that for us.

The job, admittedly, is going to be very hard, and there is no guarantee that we will succeed. Even if we commit the large sums of money that I say are going to be necessary, we still may not obtain the products which today's applications demand. What is clear, though, is that if we do not make such a commitment, we will never get there. The widening gap between offense and defense will but grow wider. So, it seems to me, we need to try. We, who collectively possess the accumulation of knowledge regarding technical methods for achieving computer security, need to apply that knowledge without the comforting constraints which we have heretofor claimed and perhaps even hidden behind. If we cannot do it, then probably no one can - and it needs to be done. There is a definite need for these technical measures.

One of the most powerful statements written in English is expressed in only ten very small words - "If it is to be, it is up to me." To apply this statement literally to the problem I have been describing would surely constitute the height of arogance on my part. Although I certainly must play an important role, it is not - it cannot be - up to me alone. So I offer a simple substitute - "If it is to be, it is up to *us*." (Slide 7) This, I believe is true. If our respective countries are to have the products they need; if we are to learn to close the defensive gap; it is largely those in this room who will have to make it happen. The security of our countries' most vital information, in a very real sense, rests in our hands. That very large computer security R&D challenge that I have been talking about is ours!

AREAS OF PRINCIPAL THRUST

- Operating systems security
- Data management security
- Network security
- Hardware and software integrity
- Formal models and verification tools

Slide 1

THE FUNCTIONAL NEED

- GENERAL PURPOSE COMPUTERS
- SPECIAL PURPOSE COMPUTERS
- SPECIALIZED SECURE SOFTWARE

Slide 2

REQUIRED STEPS BEYOND A1

- Formal proof of correspondence between TLS and model
- Verify implementation to HOL level
- Verify implementation to assembly code level
- Verify implementation to microcode level
- Verify the verifier
- Protect against hardware subversion

Slide 3

PRODUCTS BEYOND A1

	GENERAL PURPOSE				SPECIAL PURPOSE			SPECIALIZED SOFTWARE	
	SUPER	MAINFRAME	MINI	MICRO	COMM PROC	INTEL WORK STATIONS	MESSAGE SUPPORT	SECURE DBMS	SECURE COMPILER
TLS-MODEL PROOFS									
VERIFY TO HOL LEVEL									
VERIFY TO ASSEMBLY CODE LEVEL									
VERIFY TO MICROCODE LEVEL									
VERIFY THE VERIFIER									
PROTECT AGAINST HARDWARE SUBVERSION									

Slide 4

DEVELOPMENT STRATEGIES

RISK TAKER	ENVIRONMENT: UNCLASSIFIED	ENVIRONMENT: CLASSIFIED
INDUSTRY	STRATEGY 1	STRATEGY 2
GOVERNMENT	STRATEGY 3	STRATEGY 4

Slide 5

LONG-RANGE GOALS

- To narrow widening gap between offensive and defensive measures
- To better understand the COMSEC-COMPUSEC interdependencies and deal with them
- To reduce cost of computer security mechanisms

Slide 6

COMPUTER SECURITY RESEARCH AND DEVELOPMENT

Howard Weiss

DoD Computer Security Center

INTRODUCTION

This paper attempts to identify those areas of Research and Development which are presently being carried out by the Department of Defense under the aegis of the Consolidated Computer Security Program (CCSP) which is administered by the Research and Development Office of the DoD Computer Security Center.

WHAT IS THE CONSOLIDATED COMPUTER SECURITY PROGRAM?

The Consolidated Computer Security Program (CCSP) has been created to help achieve the ultimate goal of obtaining verifiably secure computer systems and networks. The CCSP sponsors generic computer security research programs which are suggested for implementation by the various services and agencies. The term generic is used to define those tasks which will have a broad application to computer security. The specific computer security research programs which are created to solve very specific problems remain funded directly by the concerned service or agency.

The generic consolidated program was begun to attempt to eliminate duplicate efforts within the computer security arena. By having a central computer security program, the roles of the various services and agencies can be more easily determined and the computer security programs being run are more visible to a wider audience.

The CCSP research and development program is divided into three major areas of concern: 1) research, 2) development, and 3) test and evaluation. The three major areas are further subdivided into subtasks: 1) security definition, 2) design concepts, 3) development and analysis techniques, 4) secure systems, 5) secure networking, 6) verification and evaluation techniques, and 7) test and evaluation.

I will now go through the subtask areas, giving some background on the types of programs that are generic to the subtask, and then try to give a feel for the types of programs that are presently running or will shortly be running this fiscal year. Because of the number of programs being run (approximately sixty tasks were proposed to the CCSP review panel and approximately 35 are being executed) the programs cannot be described in detail. Therefore, I will attempt to give you a flavor of what the various subtasks are concerned with and what results are hoped to be obtained.

SUBTASK - SECURITY DEFINITION

Background

The research in this area will be directed towards the formal descriptions of security principles such as classification downgrading, intransitive flow, data aggregation, and denial of service. The security principles will be developed in the form of formal mathematical models which will serve as the basis for formal software specification and verification. Also, under this subtask, standards and criteria will be established for the security evaluation of computer systems, networks, and add-on security packages.

Programs

There are five tasks which are being run under the security definition subtask. The MITRE Corporation will be working on the standards and criteria for system evaluation for the product evaluation groups of the DoD Computer Security Center (DoD CSC). MITRE has been working with the DoD CSC in this area for the past several years.

The DoD CSC plans to let a contract to perform an asessment of the various verification systems presently available. The work will attempt to define a process for determining how well suited the various verification systems are with respect to problems presented to them.

The Defense Intelligence Agency (DIA) will be running a program to develop a software marking system for security classification and dissemination controls which needs to be implemented throughout the DODIIS network.

Two programs will perform research in the area of formal models. One program, being run by the Air Force, will examine the area of formal models of security properties. The other program, run by the DoD CSC, will look at formal models of secure data base management systems (DBMS).

SUBTASK - DESIGN CONCEPTS

Background

The research in this subtask area is targeted towards identifying and investigating the potential of various architectures for providing secure computer processing and computer networking. The architectural concepts which will be studied include secure distributed systems, object oriented systems to support data abstraction, secure data management, capability mechanisms, and hardware support for security mechanisms. Access control measures will be studied to identify practical techniques for authentication and authorization of users and system/network components. The applicability and adaptability of encryption tchniques will be examined for data protection at various points within systems. This includes file encryption, password protection, user separation, and protected data flow. Protocols will be designed and tested to evaluate their efficiency in the secure transfer of information, both in the management of network access control and for end-to-end encryption. Designs for interfacing and integrating encryption mechanisms into computer systems and networks will also be pursued.

Programs

Several programs in this subtask are either presently running or are in the process of being let. One task will be an assessment of the capabilities mechanism architecture.

This program will examine the capabilities machine being developed under another CCSP subtask for its use as a base on which to build several different types of secure systems such as a multi-level general purpose system, a secure DBMS machine, or a secure communications processor.

The Navy will be running a program to provide for the authentication of remote terminals connected to a system. With advances in LSI technology, it is envisioned that hardware/firmware can be incorporated into terminals and hosts which will enable secure terminal authentication much in the manner of aircraft transponder identification-friend-or-foe.

A program is currently being run by the DoD CSC to study the relationships between formal verification, security fault analysis and secure architectures with repect to the design of secure microprocessor systems.

SUBTASK - DEVELOPMENT AND ANALYSIS TECHNIQUES

Background

This subtask area will address the fundamental theories and methods for producing and evaluating secure computer software and hardware. This research will be driven by requirements for both the formal verification of computer programs, system specifications and communications protocols, and the security test and evaluation of computer software and hardware. Fundamental issues that will be investigated include the applicability and extention of time related events such as concurrent processing in operating systems and communications protocols. Also, the expression and analysis of intransitive data flow as well as techniques for analyzing the security provided by the computer firmware and hardware will be investigated.

Language issues with respect to their verifiability and their ability to be used for specification, design, and implementation will be examined. The influence of language characteristics on verification will be pursued, both to direct the definition of future programming languages and to identify problems in current verification technology. Studies relating to the efficiency of verifiable languages and the use of both static and dynamic testing techniques to supplement formal verification will be pursued. Metrics for quantifying and measuring the security rating of computer systems and applications will also be examined.

Programs

The programs presently being run under this subtask include the development of a microcode verification system at the Aerospace Corporation. This work will build tools to perform the verification of machine microcode which implements machine instruction sets upon which user macrocode runs. This work is one more refinement in the verification process whereby code assurance is now being taken down to lower levels with respect to the system hardware.

The Navy is running a program which deals with the modeling of computer system risk assessment. The problem being confronted is how to formally state the security risks a system installation will face and what counter measures can be taken to assure continued system operation. Presently, there are no formal methods for making this type of assessment.

Magnetic remanence has been a problem in the past and continues to be a concern. It is unclear, even after degaussing, how much information remains on computer memory devices. A contract is presently under way to study the problem with respect to magnetic tapes.

SUBTASK - SECURE SYSTEMS

Background

The research under this subtask will be oriented towards the experimental and advanced development of secure computer systems. This research will be built upon the research carried out under the other subtasks within the CCSP which, providing results from basic research, can be applied to system development.

Formal methods will be employed in the secure systems development. As verification technology moves forward, formal models, formal specification, specification verification, and code verification will be more routinely employed.

Generic design concepts for building secure systems will be analyzed and systems of both general purpose as well as a limited function roles will be addressed. The general purpose systems are those which support such ideas as multi-level, user programmability. On the special purpose side are those systems which are not user programmable and act as secure communications front-ends, secure data base machines, or secure message systems.

Design concepts will lead towards prototype implementations of secure systems. Many of these systems will be security kernel based and will draw heavily on work performed on the KSOS and SCOMP projects. Others will be based on current work investigating capability architectures.

Programs

Under this subtask, the CCSP is presently sponsoring ten programs. Several programs are outgrowths of the multi-level security work on KSOS and SCOMP. There is work underway to enhance the PDP 11/70 KSOS system for use with the Navy's Guard project which provides a secure interface between data base systems of differing classification levels. In addition, an evaluation of the SCOMP architecture will be undertaken to see where system bottlenecks can occur and another program will look at providing a user friendly user environment on the SCOMP kernel much like a UNIX. Also, an assessment will be made of the Kernelized Virtual Machine (KVM) system which runs on IBM 370-like machine architectures.

In a continuing effort, the DoD CSC has a program to provide hardware support for tagged capabilities. This program originally started as the Provably Secure Operating System (PSOS) project but has evolved into what is now known as the Secure ADA Target (SAT) machine. On this contract, Honeywell has been working on a design architecture for a tagged machine to support a capabilities oriented system. It is also being designed with ADA as its native language.

Secure data base systems are being investigated under several programs. One Navy program will address the

building of a generic trusted computing base (TCB) to support a data base management system while another program, to be run by the DoD CSC, will address the problem of the secure relational data base management system.

Generic trusted computing bases (TCBs) will be addressed by the Navy with respect to real-time systems and to data base management systems (DBMS). Previous work in the area of building secure systems has never adequately examined either of these areas. Previous kernel based systems were anything but real-time and none have had DBMS packages on them. An interesting tying together would occur between a TCB capable of supporting a general purpose DBMS with a multi-level secure DBMS. It needs to be understood that these two entities are different - the TCB is capable of providing user, file, and process separation on a classification and category basis. But, the mechanism for separation when all the data resides in a single file and the data itself is at differing classifications and categories is a different problem that needs to be studied.

The Army, in concert with its Military Computer Family (MCF) instruction set architecture definition program, will be sponsoring a security analysis of the MCF standard instruction set to identify security flaws and to suggest security enhancements. A security specification procurement handbook is being prepared by MITRE for the DoD CSC which will supply sample security specifications for use by procurement officers.

SUBTASK - SECURE NETWORKING

Background

The efforts under this task group will address computer security issues and requirements with respect to the protection of data moving between computer systems via networks. The network environments to be examined include local area networks, long haul networks, and the internetting of various combinations of networks. Computer security issues include the development of communications protocols which support computer systems multi-level security measures and communications security mechanisms such as end-to-end encryption. End-to-end encryption implementations especially with regard to local networks and internetworks will receive major attention.

Studies will be undertaken to examine the security impact of future network technologies including distributed networks, interoperability among disparate networks, integrated COMSEC modules optical and other future communications media and techniques.

Programs

There are several important programs running in the network security area. One noteworthy program, which is actually not being funded under the CCSP but is a highly visible one which had been in its first phase, run out of the computer security research and development office, is BLACKER. The original prototype program was a success in demonstrating that end-to-end encryption across a packet switched network was indeed a viable network security solution. The new BLACKER program has just been started and the DoD CSC is playing a large role in the network security architecture.

Another program will be examining the security of local area networks. There has been an explosion of local area network products available to solve the problem of the growing march towards distributed processing in the drive to get away from central mainframes. The DoD CSC is starting in-house work in the area of studying the alternatives in securing carrier sense, multiple access (CSMA) cable bus local area networks and will be building a demonstration secure local area network.

A study contract is being let by the DoD CSC to study issues involved with internetwork security. This is a major study contract which is expected to involve multiple contractors and will try to resolve such issues as the single versus dual catenet models, security requirements for DoD protocols, and definition of security oriented protocols.

The Defense Intelligence Agency (DIA) is worried about the security of its DODIIS network and as such will run a program to enhance DODIIS security by building a set of security mechanisms which will control connections on DODIIS hosts systems. The Air Force's Rome Air Development Center (RADC) will be running a program to examine the security issues of distributed operating system protection, protocol verification, and survivability/ reconfiguration.

In order to more easily conduct internetwork experimentation, the DoD CSC is in the process of establishing an internetwork testbed consisting of a three node ARPAnet and two local area cable networks.

SUBTASK - VERIFICATION AND EVALUATION TECHNOLOGY

Background

The work under this subtask involves the growth in the state of the art of formal program verification. A major task is to take program verification from the experimental laboratory stage and move it into the operational production stage for use by people not intimately involved in verification system design or development.

Current formal verification techniques will be explored for their adaptation to verify high order languages for which they were not originally designed. Languages such as ADA and EUCLID are of particular interest. Also, techniques for software testing to supplement formal verification will be explored.

The development of automated software analysis tools for computer program evaluation falls under this subtask as well as the development of hardware analysis tools. Software evaluation tools will be developed or modified as needed to support computer security evaluations.

Programs

Programs to enhance and stabilize the Gypsy and Hierarchical Design Methodology (HDM) verification systems are being run by the DoD CSC. These programs are targeted towards making the current verification systems more user friendly and usable from the casual user perspective.

The Navy has been supporting a program with Honeywell to supply a formal top level specification (TLS), in Gypsy, of the SCOMP trusted software. The Navy is

also interested in continuing their support of the EUCLID high level language and the building of a EUCLID verification system. In the same vein, the Defense Communications Agency (DCA) is interested in using ADA to build secure systems and verifying ADA which would entail the building of an ADA verification environment. The Air Force is examining the building of an ADA formal specification environment.

SUBTASK - TEST AND EVALUATION

Background

The main thrust of efforts under this subtask are the evaluation from a security aspect, of computer products built by manufacturers and the security analysis of computer systems and networks sponsored and developed by the government.

Programs

Funding in this area has been utilized to obtain system engineering support, principally from the MITRE Corporation. The technical assistance will be used to continue to support the increasing load of commercial and government system evaluations.

SUMMARY

In this paper I have attempted to provide a flavor for the types of programs being run under the auspices of the Consolidated Computer Security Program. The goal of the CCSP is to bring secure computing into everyday use. There are many challenges that still need to be met before the era of secure computing arrives but the many programs being run are an attempt to move us along the the right path. The various subtask areas have been designed to allow the fuctional breakdown of problems which need to be solved in order to obtain the goal of building secure computing environments.

REFERENCES

1. Consolidated Computer Security Program RDT&E Detail - 1983 Compilation, DoD Computer Security Center, 1983.

2. Consolidated Computer Security Program RDT&E Detail - 1982 Compilation, DoD Computer Security Center, 1982.

3. Department of Defense Consolidated Computer Security Program RDT&E, Submissions to Technical Review Group, March, 1982.

4. Department of Defense Consolidated Computer Security Program RDT&E, Submissions to Technical Review Group, February, 1983.

DOD COMPUTER NETWORK SECURITY: PROJECTS AND PROJECTIONS

Col. John Lane

Information Systems Division, C³I

Well, when Dennis invited me to come and chat today, he said the subject was requirements, military requirements for security. I accepted fairly readily. I didn't realize the slippery slope on which I was about to embark in trying to consider requirements in general as they pertain to the military world, and security in particular. The more I thought about what exactly our military requirements were, the more I began to feel that, perhaps, military requirements in general and security in particular are a little bit like art: we all know what we like when we see it, but we're not sure we're able to articulate our likes and dislikes in advance. So we know what effective security is when we see it, but sometimes we're not able to spell out very well in advance what it is we'd like to see in systems.

The first point I'd like to make is a threshold point. And that is we can't look at security in isolation from the whole realm of, in my case, military requirements. Security, to a considerable degree, evolves from the other requirements. For example, affordability. We know that we can make our systems totally secure. We could make them so secure that the users can't get to systems to use them. It's a little bit like a bank, I suppose, where we know how to make a bank absolutely invulnerable to an armed robbery. Nobody's cracked Fort Knox yet, for example. But, if we do that, the bank's going to lose all its customers - they can't get in. And the cost of security is going to be so high, that we can't afford to put it on the system. And so, affordability becomes a driver for security.

Survivability and endurance are other military requirements that drive security. We'd like our systems to be able to survive and endure through any range of conflicts, and if we do that, it drives us into interconnection with a number of other systems. And that kind of interconnection, networking, really exacerbates the security problem. So that also impacts on what we need in the way of security.

Interoperability is another kind of an "ility" that impacts on security: to be able to interoperate in a variety of contexts, and for a variety of reasons. But the ability to interoperate means inter-netting, and that, again, makes our security problem more difficult.

I'd like to take one of these "ilities" as an example. Again, we're on the threshold of defining and trying to quantify security requirements. Interoperability is one of those requirements, and I pick it because, more than any of the other "ilities," interoperability is a driver for security. It's our need for interoperability that drives our need for effective security. By way of example of the kinds of interoperabiltiy we need, I'm going to belabor the obvious a little bit. We have to have interoperability among and between our users, and that includes the various commanders-in-chief around the world. It includes the various services and agencies within DoD, and includes interoperability with a number of other nations. There is, in addition, a kind of functional interoperability, where our intelligence systems need to interoperate with our command and control systems, which need to interoperate with our administrative and logistics systems - as another dimension to the interoperability requirement.

Again, more of the obvious: our common-user military systems have to interoperate with strategic networks, with tactical networks, with allied networks, and increasingly with local area networks. A modernization of the WWMCCS information system, for example, is being built around local area networks. Those local area networks themselves are faced with the same kind of security problem as are our global networks. When one looks at a system of systems, or a network of networks, again the security problem becomes critical.

Finally, more examples of interoperability: our common-user packet networks need to operate with the networks of our allies. They need to interoperate with the PTT networks. They need to interoperate with the packet radios, the packet satellites, and local area networks of all sorts, sizes, and descriptions. If we look at the military requirements mix, they drive us in the direction of internetworking, a network of networks. Virtually every "ility" that we hold dear in the military demands robustness, and redundancy, and flexibility in our networks, and it all pushes us toward an inter-network architecture. So it is in that context of an inter - network architecture in data communications that we need to concern ourselves with the security requirement.

The people on this slide are highlighted in white to indicate that our security is no more effective than the trustworthiness of personnel who operate and maintain our systems. But the point behind the whole slide is that we have a range of areas which are vulnerable. We have hardware, firmware, I guess we can call the people "humanware" eventually. (I know there's a social scientist out there who just died in his seat when I said that.) We have a range of vulnerabilities in the network and in the ADP. We need computer security. We need communications security. We have a dimension of physical security, personnel security, emanations security. The point here is that computer security is only one piece. We have to have an overall balance. It does us little good, as an example, looking at emanations security, to attempt to protect an entire computer facility, if we're pushing the data out on unclear lines that can be readily intercepted and tapped. So we have to have a balance. In any aspect, security is not an absolute; it's a balance. We have to balance it not only with other military requirements, but also, in another context, across the range of all our vulnerabilities. Again, that slide says you've got to have a balanced approach to security.

What are the security-relevant considerations or factors in the kind of network we're looking at? I don't know whether to call these "requirements" or "considerations" or

"characteristics" of the network. I found that I had a terrible problem when I was trying to categorize these things; they all seem to flow one into the other, and didn't really fall into a niche. But let me just call them all "considerations" that impact on network security. One of those is the traffic that actually flows. In the kind of networks that we're faced with in the Defense Department, we have all sorts and kinds of information, ranging from unclassified (unclassified as a single message, but, perhaps in an aggregate sense, we begin to approach classified information, if we have enough of that unclassified information) to classified information of all sorts and kinds, ranging from CONFIDENTIAL up through the highest levels of compartmented information. And we also have a considerable amount of overhead traffic in the network, which may be important security-wise, because it may give away a lot about how the network functions, how it's managed, and what the state of the network is at any given time. We're concerned about traffic flow security. You all know what traffic flow security is, but I'll tell you anyway: the form of the information that flows in the network that has nothing to do with the content of the information, such as addressing information or the raw quantity of information flowing between any two points. Someone who is looking can glean certain forms of intelligence from this. So we have to protect the addressing information, as well as the information concerning the quantity that is flowing in the network, all under the rubric of Traffic Flow Security.

Privacy. We share with the commercial world and with the federal non-military world privacy concerns under the privacy legislation. Personnel information, medical information - all of that kind of information must be protected, although it's not classified national security information. There's also information that flows that's so highly sensitive, only a designated individual should read it. We have compartmentation, which is a super kind of a need-to-know requirement, where the information is so highly sensitive that only a small subset of people who are specially cleared are allowed to see it. Then we have the general need-to-know requirement, where we would like to be sure that someone who looks at a particular piece of classfied information really has a need to have access to that particular piece of information, even though he may be cleared to the proper security level. Increasingly, in a network environment we are faced with multi-level secure hosts. There are not many today, a few. But a number of programs coming down the road, which will be providing multi-level secure hosts, will be expecting to talk into our data communications network. Networks have to be able to provide a degree of security commensurate with, and which will protect, the various levels of information which are passed into the network.

Then, user expectation. Again, this probably could be in the category of requirements. Some users expect to have an originating terminal verified. They'd like to know from exactly which terminal the particular message came, or from which particular user a particular message came. We need protection against spoofing, or the injection of various traffic which might appear to direct some action. In general, the users demand high reliability on the network. They demand affordable security. They demand a trustworthy network: in the record communications world, for example: no interlacing of messages, no stragglers.

If we boil all this down, we approach what we frequently see as a classic statement of the security requirement. If we are going to protect information, we want to provide traffic flow security. We want to provide anti-spoof protection, anti-denial-of-service protection, and provide community-of-interest protection. I think the solution to all this was said very well yesterday by Dr. DeLauer; as he concluded, we have a long way to go. We've embarked on that path.

Our evolutionary approach to this solution is the Defense Data Network, which is an evolutionary and common-user data communications program which folds together a number of networks in an inter-networking environment. It's very interesting that security was the primary architectural driver of the Defense Data Network. I think that I can illustrate that very well with another slide that I borrowed from the Defense Communications Agency. I think there are a couple of important things about this slide. This is a slide that the Defense Data Network program manager uses to give overviews of the network and to give program status reviews. I'd like to point out that, if we look across the top of this chart at the major milestones, those which the program manager considers totally significant, we find that the availability of an internet private line interface, or an end-to-end encryption device, is an extremely significant event in his eyes, as is the obtaining of the second source for those, so that those units can be produced in quantity. Farther on, the availability of the BLACKER program, which is a very sophisticated end-to-end encryption program, is highlighted. If you'll look down these lines that fold in great profusion across this slide, you'll note that you see phrases: SECRET, TOP SECRET, SCI, TS. Security is the major architectural driver. Looking at the bottom of the network, you'll see the ARPAnet you're all familiar with. The ARPAnet we split on 4 October into an experimental network for our network researchers and an unclassified segment. The MILNET is an unclassified network. The MILNET and the ARPAnet are connected through a mailbox gateway.

Now, above this line, in red, are all the various classified networks, networks that handle classified information. At this point in time, we have or are building separate networks for each security level. The reason for that is that we don't have a device today that we can use in quantity that would allow us to mix TOP SECRET, SECRET, and unclassified information on the same network. That's the IPLI, or Internet Private Line device. We have a similar device on the ARPAnet in use today, the private line interface, an end-to-end encryption device which uses older technology and does not have an internet protocol. The IPLI is a device that will become available in early '85 in limited quantities, which includes an internet protocol that will allow us to operate in an inter-network environment, and also is a cheaper version. At the point when those devices become available in quantity, we will begin to fold these separate networks into a single classified network. BLACKER will give us the capability to dynamically key on a message-by-message basis, so we get true interoperability between the various communities of interest.

Another DCA slide, here, which shows in a notional way the split between the classified segment of the network and the unclassified segment. On the classified segment,

we'll be protecting the access lines with the IPLI end-to-end encryption device where that's necessary to preserve a community of interest. The trunks will use standard key generation equipment to protect the trunk, so that much of the information flowing on the trunk will, in fact, be super-encrypted. On the unclassified side, we intend encrypting the trunks with DES, and the access lines can optionally be equipped with DES when that's necessary. What this chart doesn't show is another segment of the network, which is what we today term the ARPAnet - that portion of the network used by the network researchers.

Let me shift gears now. I tried to talk in some way about military requirements. How do they compare with commercial requirements, civilian requirements, or the non-military federal side? Well, if you look at the applications, you can match the sets almost exactly. Almost any application you can think of in the military is used in the civilian world, and vice versa, even to intelligence applications. So, the difference is certainly not in applications. If you look at techniques that we use or should use to protect and secure computer systems and networks, the techniques are the same. So it's certainly not in techniques. The differences in security, I think, are not qualitative; they're quantitative between the military and other worlds. By that I mean, the penalty of failure of security is much more severe in the military world, where national security, national survival at the extreme could be at stake. One could compare that to corporate survival if the wrong information about a marketing strategy got out; but in terms of impact to our society, certainly not as extreme.

Another area where I discern a difference, although I think this is arguable, is in the need for interoperability. I believe in the military we have greater needs for interoperability with other systems than a typical user does in the commercial world. This, it appears to me, drives more stringent security requirements for the military. I think the two of these together, in their practical effects, mean that in the military we're probably willing to spend more on security than a typical non-military user is willing to spend. But it also appears to me that the trend is changing. I think the publicity that's been given to hackers has done us a lot of good in the computer security community, because they've focused attention on what we've known all along were our vulnerabilities and our weaknesses. They've caused us to apply a lot more attention - maybe not have caused *us* to apply a lot more attention, but it caused the people we work for to become much more aware of the computer security problem. As a result, greater resources are going to be put into this area.

What's the bottom line? I think the message that I'd like to convey is to follow up on what Dr. DeLauer said yesterday. DoD is increasingly serious about network and ADP security. Dr. DeLauer mentioned that there's an old saw in the industry that a computer procurement has never been won or lost on the basis of security features offered. Well, it's our intent in the Department of Defense to turn that around, and to make computer security features a very important part of procurement. We intend before next year is out to have in place mandatory directions to all the services and agencies to specify in their ADP procurements the requisite degree of security, based upon the Computer Security Evaluation Criteria document that has now been published by the DoD Computer Security Center. If a particular system has no requirement for security - that is, if the information doesn't deserve any protection at all - we intend that the program manager make the conscious decision that it requires no protection and specify a D-level of protection, and so on.

I believe that the message to the vendors ought to be pretty clear on this. Affordable computer security *is* going to sell, and we all need to start marching along together to provide security features in commercial offerings. One only needs to look at the trade magazines now. In virtually every trade magazine that comes out, there's an article or a page or an editorial devoted to the subject of computer security features. I think the handwriting is on the wall. We're doubly serious about computer security, and I hope that the commercial world reads that message and starts marching along in the same direction.

Question and Answer Session

Q. Perhaps it would be a bit of an elaboration on the one point that you mentioned about next year starting to require the trusted Criteria in your specifications. We have been pursuing various programs, where we go in and talk with the program managers or the related personnel, and start mentioning, well, what is your requirement for security, what type of formal specification/verification do you want? As soon as we start mentioning those sorts of things, it reverts back to, well, we better look at this system-high type of operation for right now because of the newness of the technology. Could you elaborate on that a little more - on your plans for incorporating this down into the ranks?

A. Yes, right now there is a memorandum flowing to Dr. DeLauer's office for signature to the services and agencies, which releases the Computer Security Evaluation Criteria, and asks, pending formal coordination with the services and agencies, that those Criteria be taken and used in ADP procurement. What we hope will happen is that the ADP managers in the services and agencies will get very closely involved with the personnel at the Computer Security Center, so that they can understand the kind of technology that is becoming available in the commercial world, what it will do for them, and that they will be able to better articulate the levels of security they require in terms of the Criteria.

Q. I was curious about your statement about using the Criteria in requiring networks - are you planning on requiring an A 1 network? Or are you just going to make requirements for A 1 systems within your network?

A. Let me say first that I forgot to give my caveat at the beginning of this session that I wasn't going to foist myself off on you folks as a computer security expert. So please don't hold me to precision in my use of terms. In building the Defense Data Network, we're going to rely a great deal on end-to-end encryption. Thus, if the Evaluation Criteria could be applied to a network as opposed to a particular product, the network would reach a high level. We have to be able to maintain whatever degree of security is required by any system that plays into that network. For example, we expect the inter-service agency AMPE development will attain an A1 level. The AMPE will provide our DoD common-user record communication

system, which will use the DDN as a community of interest. Since that system will be secure to an A1 level, the network will have to protect to that level. I have a hard time thinking in terms of the Criteria applied to a network as opposed to a product.

Q. Basically, the question was, if you are planning on doing that, or if you were going to wait until the Center comes out with a network document, and try to apply that. I was curious which way your plan was going to go, saying, do we want A 1 systems in our network, or do we want an evaluated network itself?

A. Well, the pieces of the network, for example, the IPLI devices, as a piece of ADP, require a certain Evaluation Criteria level that is pretty high. Various pieces of the network require it. But then, you have to look at it as a whole. We're looking at end-to-end encryption as being a major protector of information on the network. That reduces the requirement on the network itself. If we have all black switches in the network, then those black switches will require a lesser degree of security. Have I talked around that question sufficiently?

INTEROPERABILITY

SECURITY

SURVIVABILITY
(ENDURANCE)

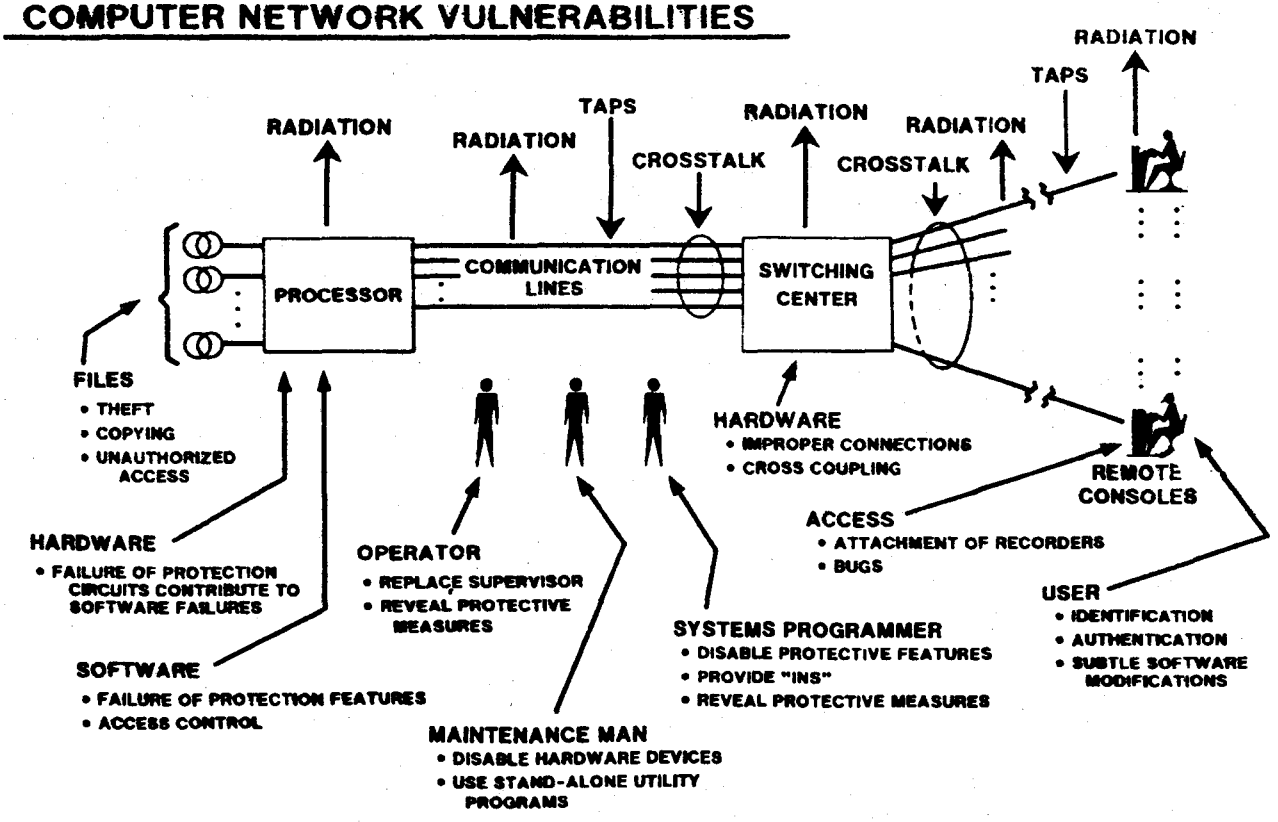
AFFORDABILITY

Security Requirements Cannot be
Considered in Splendid Isolation

Symbiotic Relationships in the Requirements Matrix

Slide 1

COMPUTER NETWORK VULNERABILITIES



Slide 2

SECURITY RELEVANT CONSIDERATIONS

TRAFFIC - a mix

- unclassified

- classified

- overhead

TRAFFIC FLOW SECURITY - important for certain communities of interest

PRIVACY -

- personnel records

- highly sensitive "eyes only"

COMPARTMENTATION - within a classification level

"NEED-TO-KNOW" PROTECTION

MLS HOSTS

USER EXPECTATIONS

Slide 3

THE SOLUTION

"Our ideal information system would allow totally secure simultaneous use of a processor for all levels of classification from unclassified through the most sensitive information, and transmittal of that information through a network securely accessed by multiple users at different security levels. I mean, of course, a truly multi-level secure processor operating into a truly multi-secure network, with ease of information interchange between and among users at all security levels. We have a long way to go."

Dr. DeLauer

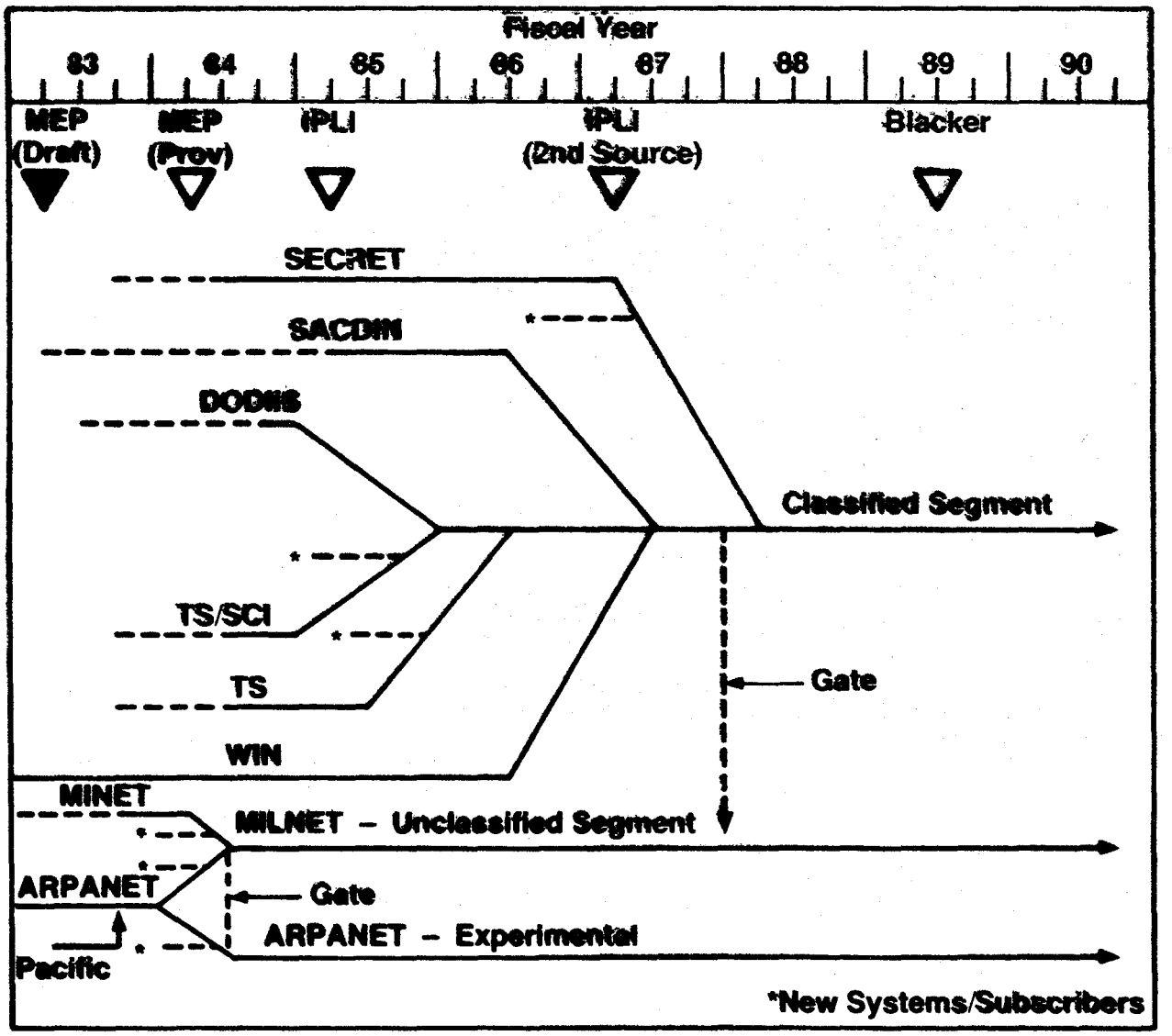
November 15, 1983

Slide 4

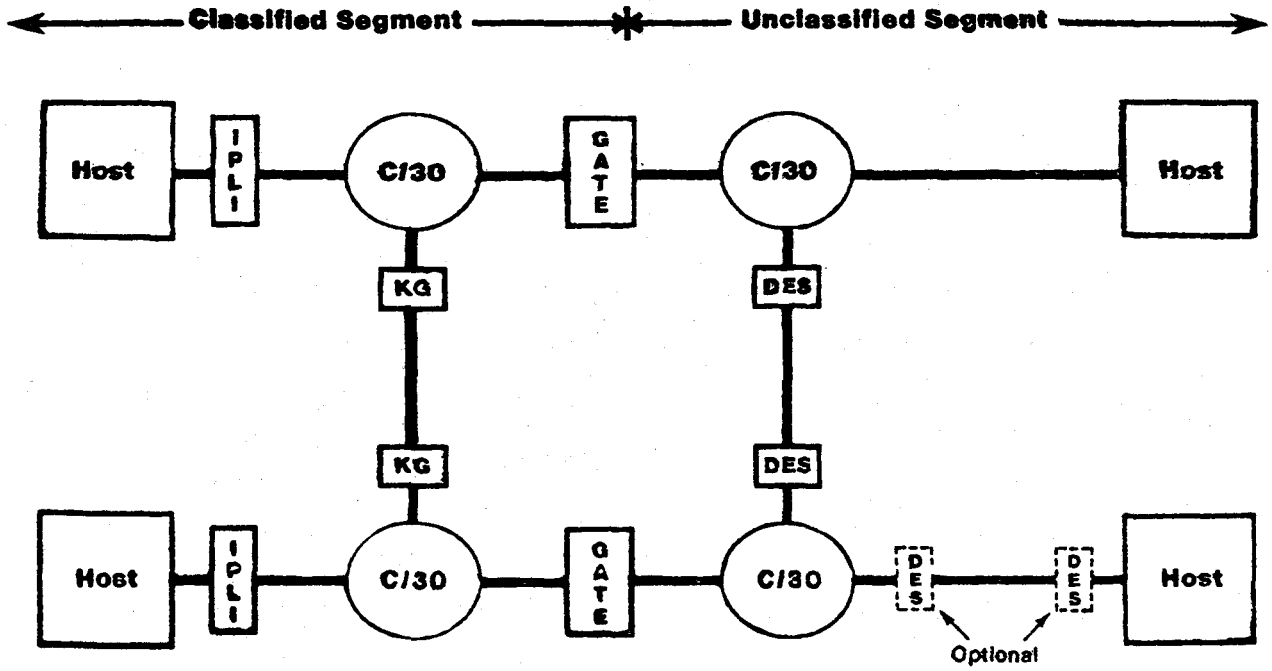
DEFENSE DATA NETWORK

- Evolutionary and common-user data comm program
  - Internetwork architecture
    - Security - a primary architectural driver

Slide 5



Slide 6



Slide 7

THE MESSAGE

DOD IS SERIOUS ABOUT NETWORK/ADP SECURITY

- mandating requirements in all networks for specified security level

- affordable security will sell in military and non-military sectors

Slide 8

COMPUTER NETWORK SECURITY: PUBLIC AND PRIVATE

Dr. Stephen Kent

BBN,Inc.

For the next few minutes, I'd like to talk about a system designed to deal with a particular kind of network access control problem. It's especially appropriate following Col. Lanes' talk and the subject of this session. Not only is it an example of a system which is based on experience with prototype systems in the network security environment - although we didn't build this specific system in the laboratory first - it is a system which, in a slightly different form, is about to be deployed in one of the networks that Col. Lane was just talking about, a portion of the Defense Data Network; and it is, in fact, already deployed in a fairly large commercial packet switching network that we provided. So it's a perfect example of where the military and commercial requirements for network security - in this case, for access control - converge quite closely, and one is able to provide essentially the same sort of product to both. You'll notice that my slides are, in fact, even less elaborate than Col. Lane's, because after the four and a half billion dollars has been filtered through various agencies of the government down to the contractors, we have to be very frugal in spending it. So mine are done just in terms of your basic black-on-clear.

What we'll be talking about is not actually all that new. I like to think of it as just another saga in the continuing struggle between good and evil. We are, obviously, the good guys, or we wouldn't be attending this symposium. What we're going to look at today is a network access control system designed to deal with a particular kind of access control problem. We'll examine it in its military substantiation, where it's called TACACS: the TAC-access control system, because all of us associated with DoD have a real fondness for acronyms, and TAC is the acronym for a Terminal Access Computer. It's a mini-computer used to provide access for terminals to packet-switch networks of the ARPAnet flavor. This system, which will be implemented - we're in the process of deploying it now that we've tested significant portions of it - is going to be used in the MILNET, the unclassified segment of the Defense Data Network, based on the ARPAnet packet-switching technology. As Col. Lane pointed out, the first phase of the split has already taken place to divide the old ARPAnet into two pieces. The second phase, to make that a rather permanent split, and a much more physical split, will be taking place early in 1984. This system will be going into effect in that time frame.

The TAC is a system that provides access for up to 64 terminals. These terminals may be hardwired or may come in over telecommunications lines. Typically, the telecommunications lines are dial-up lines, and therein, of course, lies the primary problem. So, the mission of TAC-ACS is to control access to the TAC resources, primarily, to the dial-up resources - the dial-up ports on the TAC's. One way of looking at its goal, its purpose, is to reserve these resources for the good guys, to keep the scum of the earth off the TAC ports. The threat comes, if we look at the broad spectrum of bad guys out there, from the KGB on one end to, you know, Suzie Thunder and the 414 Club at the other end - I believe they're at the other end. We're trying to deal with the folks toward the Suzie Thunder and 414 Club end of the spectrum. So this kind of system will not, in fact, offer protection against wire-tapping attacks. We're just concerned about keeping people with terminals and personal computers from gaining access to the network and to network resources. There's a secondary benefit from this. That is, if we keep unauthorized users from accessing the network in this fashion, then, in a sense, we're providing a sort of outer perimeter of security for hosts on the network, in terms of those host's being accessed through the network itself. Now, like any fence, there are some holes in it, and the holes exist in this case because many of the hosts on the net have dial-up capabilities directly, and of course, that's beyond the purview of the network security mechanism for the backbone net. If someone were to come in through those facilities, because of poor security controls on the host, and then get into the network, there's not a lot that we can do about it, and this system won't address it. However, this *is* designed to close some of the larger gaping holes that exist today.

Let me mention a little bit more about the requirements. The goal, of course, is to keep the bad guys out. However, there is also a goal of letting all the good guys in. For those of you who are fond of statistics, we're trying to really minimize the type 1 and type 2 errors in this system. Therefore, a very high degree of availability is required of this particular network access control system. It's not acceptable to say, "I can't tell the difference between good guys and bad guys at the moment, so nobody gets to come on." That's not an okay thing to do, since we are making life more difficult for the good guys as well, because we're requiring them to do something else before they can get their work done. We should try to be as user-friendly as possible, and we should try to minimize the unpleasantness associated with this, and make it all go by pretty quickly.

Finally, there's an audit trail requirement with this system. The need for the audit trail here deals with several things. One thing: it's nice to know in a network how often people are really dialing up and using the systems, and it's convenient to tack on this kind of collection of information to the functions that already have to take place as part of user authentication. However, as we'll see in a system like this, it's important to detect when passwords and authentication strings have been compromised. We expect some users would actually let us know if they think that their passwords have been lost or stolen. But there's always this possibility: that in the very large community we serve, some of the good guys may not really think that this kind of an access control system is the right thing to do. They might have a religious objection to it. And they may, therefore, tend to *lend* their passwords, make them available to some of those folks who fall into the scum-of-the-earth category. Since they're not going to tell us they did this

kind of lending, it's up to us to detect it. One way that we can have a good shot at detecting it is to maintain audit trail records and do non-real time analysis of these records to detect patterns of use which are indicative of something being wrong out there. For example, MILNET is a large network covering the span of the continental United States and some outlying posts, as well. If, in analyzing audit trail records, we find the same user logs in from a TAC on the east coast, and within ten or fifteen minutes, appears to be logging in from a TAC in Tegu, South Korea, we have reason to suspect that something funny is going on here. If it's not a compromise of password, we want to know how he gets between these two places that quickly! So, one way or another, we want to follow up on that.

The first phase of the TACACS system is based on the concept of a self-validating log-in string, which is illustrated here. It's in two parts. The user name is a unique identifier for the user, and following that is something that the user views as his password, which is nine characters in length. In fact, two of those characters are not part of the validating portion of the string; they're tacked on to the user ID to make it unique over time, because users will have a tendency, as we all know, to lose passwords - they'll wind up being disclosed, something terrible will happen. We'd like you not to have to change your name over time; that should remain constant. Yet the principle behind the self-validating log-in string is that we take the user's name and transform it in some complex fashion to yield a log-in string: to yield the check-character portion of the log-in string. So, if you were to lose the check-character part, the password, then how do we issue you a new one without changing your name? Well, the answer is we just increment your user-version specifier, which is an indication of how many times you've screwed up and lost this thing. This provides a very simple verification mechanism. Notice we don't need a table of all the good guys, which is the reason we're doing this first. All we have to do is take the name, the user and group version numbers (I'll mention why group version number is important later), transform them, take the results of the transformation, check it against the check characters (cleverly named); and if they match, then this is a valid pair. For those of you who are crypto-junkies, we use, in deference to our host, the Data Encryption Standard in cipher block chaining mode with a zero initialization vector and a secret key. Wasn't that fun? And out of the 64 approximately random bits that come out of that, we map these down into seven check characters, which are drawn from an alphabet of 32 characters each, so there are 35 bits of verification information, which, if you can do some quick arithmetic, is a bunch. The problem with this scheme is that people will lose their passwords, or have them disclosed in some fashion. We have to be able to deal with it. We can't tell the difference, in terms of the check function, between a disclosed, formerly good password and name combination, and a currently valid one. Therefore, there needs to be a "hot list." That's one of the less fun parts of the system, but there is some cleverness in that, too.

To understand how the system works as a whole, since we've looked at detailed mechanism of the log-in string, we need to see what the other parts are and then look at a diagram that shows where all the bits flow. The validation is performed in the initial system by what's called the fake host in the IMPs.

The IMPs are the packet switches in these networks. We are able, without a tremendous amount of effort, to provide a DES capability in these packet switches. It doesn't have to appear in all of them, it just has to appear in enough of them so that the ability of a TAC to reach one of them is very high. Since the network does not work if there are no packet switches operating, this is a reasonable baseline. That is, if they're working, then certainly, you have this verification function, and you can talk to other people. If none of the packet switches in the network were accessible, we wouldn't care who you were because there would be no resources to access. Seems like a reasonable marriage. The TAC acts as the access control point, gating the user on the system. It does the hot list cheking. The packet switches already have enough to do. We're asking them to do this tiny little bit of verification, but we don't want to burden them with holding onto hot lists. The TAC will also send the audit trail information that it collects, since it's the user's point of access in the network, to a user data base host.

There's one user data base host in the system. It does not have to be available on a continuous basis because it is only the clearinghouse for getting new passwords, new log-in strings, reporting old ones, which are human functions - you call people up, let them know about this. That doesn't require a machine to be working all the time. It provides the centralized database that has all this good information on it, and maintains the hot list - the master copy of it - and actually distributes it through another component of the system.

What's important, in terms of continuous availability, are the authentication servers, which are in the packet switches themselves. The flow of information in the system is shown here. The TAC accepts the user's name and his password, packages it, and sends it off to a verification host, which, in this case is an IMP, a packet switch. It tries to use the one that it's attached to; it tries to use that because that's the closest one. As long as that one is equipped with the necessary DES keys, it can go ahead and do that. We may not put these keys in all the packet switches, because some of them may be located in sites which are considered to be a little less secure than others. So, we might not put them there. But each TAC has a list of IMPs that it *can* go to for performing this function. After sending off the request, in parallel the TAC searches its hot list to see if this particular log-in string has been compromised recently or at some time in the past - where it is in the version space of the given log-in string. When it gets a response back from the verification host, it checks it. If the verification host says, "Yes, this is a good log-in string," and if you did not find this log-in string on the hot list, then the user is a good guy and he's allowed to use the resources of the network. After he logs in once, he can continue to use these resources until such time as he actually hangs up and disconnects, turns off his terminal, or issues an explicit log-out command, This means he can open a connection to a host, do some work, close the connection, open a connection to another host, do his work, and be inconvenienced only once at the beginning, as long as we have confidence that he's still there.

A secondary function of this system is that the TAC sends the audit trail information to the user data base host up here for later analysis, and so we can find out just how many users are out there in the system. The other path is from the user data base host updates to the hot list, which are sent to the network monitoring center, running the Network Utility program indicated here. That network monitoring center will send the updates to each of the TACs in the system on a daily basis. Since the network monitoring center already converses with these TACs to down-line load their code or changes - make changes to the configuration of the software for each of these TACs on an individual basis - it's in the best position to go ahead and perform this function. So that's the data flow in the system. The hot list, for those computer science junkies in the audience, is a tree structure of depth either 2 or 3, depending on whether you're the kind of person who counts from zero or one - I believe that's the distinction between computer scientists and electrical engineers. The tree structure occurs because a user's name, for instance, in our earlier example "GParkerDDN" is a two-component thing in which part is the user's name, and the second portion is a group with which he is affiliated, perhaps a host on the network.

In order to make the management of the the hot list feasible, user groups can be wiped out as a whole. That is, we can reissue new passwords for an entire group. Because TAC hot list space is limited, if we were to overflow we would not be able to tell good log-in strings from bad log-in strings, that is, formerly disclosed ones. We have to make sure it doesn't overflow. One way to do this would be to reissue passwords to everyone. That's painful. It's good for the U.S. Postal System, but it's painful. So, what we're capable of doing is reissuing passwords to subsets of the subscriber population. We can do that by seeing which subtree has grown large and say, "Aha! You guys have been messing up a lot and losing your passwords. Guess what. You're all going to get new ones." Then we get to delete that subtree here. That provides a more manageable way of dealing with the problem. It also offers another important advantage in that it allows us to have guest accounts, so that host liaisons can be given pre-printed forms with dummy user IDs and log-in strings to hand out to people who walk up, who are established as good guys, and who want to get on the system now. Since we all know the delays bureaucracies introduce into systems, this is a way of letting users go ahead and use the system quickly while we process all the paperwork to actually make this permanent. However, we want a limited lifetime on these guest accounts. So one issues the guest accounts under particular groups, which are periodically wiped out. We up the group version number, therefore wiping out all the old ones. That's a way of managing this problem as well.

The second phase of the system makes a couple of minor changes. The primary change is that instead of going with this distributed authentication function by distributing it throughout all the packet switches, we'll provide some number of what are called "login hosts," which will be MC 68,ØØØ-based small hosts, stand-alone, no-human-intervention-involved hosts, scattered geographically around the network. The queries will be directed to them, instead of going directly to selected packet switches. Now, the good news is that, by directing the queries to these hosts, these hosts can maintain not just the simple function of "are you a good guy or not," they can actually maintain copies of a compressed form of the data base with all the good guys in it. So, we'll be doing a positive check. And that means we won't have to maintain the hot lists distributed in the TACs anymore, and that is an improvement.

Notice that when one goes to this sort of system, it's actually conceivable that you could go to a different type of password scheme which is no longer self-authenticating. You could have separate names and passwords where the passwords were selected by users, but subject to extensive screening to avoid user names, commonly used words, names that are too short. You can enforce periodic changing, etc., or you could go to system-generated random password strings that were pronounceable - those sorts of things. There's a chance we might go to that in the future. However, since availability is such a critical concern in a system, we won't go to that in the near term because the IMPs with the fake hosts still provide a backup capability, should for some reason all of the login hosts in the system suddenly become unavailable. Now, if experience shows that they are unlikely to all become suddenly unavailable, except in the case of major war - in which case it's not clear how important some aspects of this are. As long as the postal service is going to deliver hard-copy mail in a post-nuclear war situation, we should be prepared to deliver electronic mail; it seems only fair. It might be possible with experience in the availability of the log-in service to say this backup capability isn't critical. But this is the way the system will look in its second phase, where the TACs will send out a validation request to a login host. In fact, they can't tell that this has changed in any fashion - it's just a different address for the log-in host - and they'll get a response back, and their life will be simpler because there will be a null hot list to search. So it's completely upward compatible as far as the TACs are concerned. The IMPs will only be used for backup purposes. The audit trail information will still be sent to the data base host, retaining its function. And the data base host will now send updates, new users and cancelling old user entries because of loss or inadvertent - whatever sort of disclosure - directly to the login hosts, which are replecated in small numbers throughout the system. This is closer in flavor to the commercial system that is in place now. The commercial system makes similar use of the Data Encryption Standard that we described earlier. The primary difference, of course, is where you get the DES keys. One uses a genuine COMSEC quarter to do them for a DES system in the military environment, or you can use any old proof coin from the Treasury if you're just doing it in the commercial environment.

Thank you.

SECURITY MECHANISMS IN LINCS

D.M. Nessett

Lawrence Livermore Laboratory

Abstract: *LINCS is a distributed operating system currently being implemented at Lawrence Livermore National Laboratory to support multi-security level, multi-compartment distributed applications. Aspects of its architecture that relate to distributed system security are briefly discussed.*

1. Introduction

LINCS (Livermore Interactive Network Communications System) is a distributed operating system currently being implemented at Lawrence Livermore National Laboratory (LLNL). It has been specifically designed to support distributed applications running in a classified environment. Applications running at different security levels and processing the information of separate classified programs *(compartments)* are to be multiplexed by LINCS over a large set of heterogeneous computational resources.

2. LINCS - An Object-Oriented Distributed Operating System

Normally when one thinks of a distributed system, its physical topology comes to mind (fig 1). Hosts (called here, distributed system nodes) are interconnected into local area networks (LANs) which are themselves interconnected by a communication subsystem to form a complex amalgamation of computational resources.

In general, the various LANs of the distributed system are based on different networking architectures and implementations. For example, in fig. 1 two networks based on proprietary software (DECNET) are shown interconnected with an in-house network and a remote network each based on their own individual architecture and implemented by locally produced software.

Each node in the distributed system multiplexes its resources over one or more computational units called *processes.* These processes cooperate and communicate with other processes in the distributed system to form distributed applications.

While this physical viewpoint is proper when thinking about certain aspects of a distributed system (e.g., maintenance of the interconnection equipment, some aspects of distributed system security - see below), when thinking about how to structure services in a distributed system, another viewpoint is preferable (fig. 2). This viewpoint emphasizes the logical structure of the distributed system and focuses on the processes that execute in distributed system nodes as well as on distributed interprocess communication.

LINCS [WAFL79,FLWA82] is best described using the logical viewpoint. Services provided by each LINCS node operating system including peripheral device servicing, process scheduling and resource mutiplexing, protection and security, and interprocess communication coalesce to form a *distributed operating system kernel.* Processes in the distributed system interact by using the distributed interprocess communication facility provided in the distributed operating system kernel. A process possesses one or more *ports.* Communication between two processes occurs over an *association* which is a reliable interprocess communication channel formed by the distributed operating system kernel between a pair of ports.

The operating system of each LINCS node also supports an interface between its services and the processes in the distributed system providing them with a coherent and uniform view of distributed system services. The interface is implemented by system processes called *servers.* A server arbitrates access to a set of low-level system resources (e.g., disks, magnetic tape, main memory, processor cycles) in such a way as to give accessing customer processes the view that they are dealing with high-level abstract objects (e.g., files, processes). Servers also utilize existing abstract objects (e.g., files) to create other types of abstract objects (e.g., directories). The amalgamation of these server implemented interfaces together with the distributed operating system kernel is called the *distributed operating system.*

Each server supports a *service protocol* by which its objects are accessed and manipulated. Objects of the same type (e.g., files) are accessed by a standard LINCS service protocol associated with that type of object (e.g., the LINCS file-server protocol) no matter which of a number of different servers managing objects of that type is being dealt with.

One of the most important aspects common to all LINCS service protocols is the way in which access to LINCS objects is controlled. A customer process is allowed access to a particular object only after it has presented to the object's server a protected name of the object. This protected name, called a *capability*, contains a set of access rights (e.g., read access, write access) that allows only certain types of operations to be performed on the object the capability identifies. The capability is protected so that it can neither be forged nor stolen and then used *(useful theft).*

3. Fundamental Principles of Distributed System Security

While the logical viewpoint of a distributed system is superior to the physical viewpoint when designing a standard set of distributed operating system services, an analysis of the security posture of a given distributed system requires careful attention to its physical structure [NESD83]. Designing security mechanisms for a distributed operating system such as LINCS, therefore, must be driven by both viewpoints.

The aspects of a distributed system's physical structure relevant to its security can be summarized in a number of fundamental principles. These principles are applicable to a general class of distributed systems and distributed system architectures. That is, although the security mechanisms in

LINCS are based on these principles, they are quite general in scope and are useful in the design of security mechanisms for a large variety of distributed system architectures.

3.1 First Principle - The Parts of a Distributed System Cannot be Trusted to the Same Degree [WAFL79, WATR8Ø]

This principle of distributed system security is based on a number of observations. Firstly, the nodes and LANs *(distributed system parts)* that form the distributed system will, in general, be administered by more than one authority (fig. 3). Certain authorities may trust the parts administered by other authorities only to a limited degree. In addition, some authorities may impose stricter or substantively different physical security controls on their nodes and LANs than other authorities leading to different vulnerabilities in separately administered distributed system parts.

Secondly, even within the same administration, different distributed system parts may be located in areas with quite different physical security controls. Physical access to some areas may be limited to only a small group of personnel with common need-to-know requirements. Other areas may have no physical access controls and be interconnected to terminals, nodes and LANs through public telecommunications equipment.

Finally, in general the nodes of a distributed system will run different operating systems possessing various degrees of trustworthiness. Some operating systems may have known vulnerabilities. Certain operating systems may have been closely scrutinized and possess a high level of trustworthiness. Nodes that run operating systems with known or potential vulnerabilities will be called *vulnerable nodes.* LANs built up by interconnecting vulnerable nodes will be called *vulnerable LANs.*

3.2 Second Principle - The Foundation of Security in a Distributed System is Secure Interprocess Communication [WAFL79, WATR8Ø]

The security of interprocess communication is fundamental to distributed system security for a number of reasons. Firstly, the origin address of a message is used many times to enforce other higher-level security constraints. For example, the capability protection mechanisms mentioned below [DOFL8Ø, NESD82] require that the origin address of a mesage containing a capability be correct. If, for security reasons, a group of physically-isolated vulnerable nodes or LANs is treated as a single entity, it may be necessary to guarantee the origin address of packets only to a level of granularity necessary to identify that group.

Secondly, the distributed interprocess communication facility must ensure that the data it moves is properly handled according to the data's security level. This normally requires that all packets moved by the distributed system's communication subsystem be labeled and handled in such a way that various interprocess communication threats [KENS8Ø, NESD83] are properly thwarted.

Both of the above requirements can be met by a combination of trusted hardware/software (for example, trusted LAN gateways) and end-to-end encryption techniques (fig. 4). Trusted gateways can ensure that the origin address and security level of packets arriving from other areas in the distributed system are within reasonable bounds by observing which specific communication channel is used to move them into the gateway. Gateways can establish trusted communication channels between themselves and other gateways over untrusted communications equipment by using end-to-end encryption.

3.3 Third Principle - Distributed Systems Can be Compartmentalized by Using Both Trusted Components and Vulnerable Components Protected by End-to-End Encryption

For economic reasons it is often desirable to utilize as much commercially available equipment as possible in a general purpose computational facility [WATR78]. This desirability generates a requirement to process classified information on vulnerable nodes and LANs. Processing classified information on vulnerable nodes or LANs can be allowed only if: 1) physical access to these nodes, LANs and their peripherals is strictly controlled, 2) the information processed on these nodes and LANs is from exactly one compartment, and 3) all personnel allowed access to these nodes, LANs and their peripherals are cleared to the highest security level of information processed and possess the proper need-to-know requirements.

Interconnecting these physically-isolated vulnerable distributed system parts to trusted parts can be achieved by the use of end-to-end encryption (fig. 5). In the past this approach has been suggested for the protection of communications between individual nodes [HHDU82]. However, the approach can also be used to protect communications between vulnerable LANs. In particular, if a trusted end-to-end encryption device is attached to vulnerable LAN gateways and if all communications to/from the LAN are forwarded through these gateways, access to vulnerable LANs can be properly controlled even though LANs in different compartments utilize a common communication subsystem. Physically separate vulnerable LANs in the same compartment are able to communicate over an untrusted communications subsystem by keying their end-to-end encryption devices with the same keying material. LANs in one compartment are not able to communicate with LANs in other compartments because their end-to-end encryption devices are keyed with different keying material. Trusted LANs are able to communicate with LANs in different compartments by possessing keying material for LANs in more than one compartment.

4. Non-Discretionary Security in LINCS

Currently, LLNL's Octopus Network supports multi-level, multi-compartment computation. As LINCS is integrated into the Octopus Network, this type of processing is expected to continue. The Octopus Network part of the LINCS distributed operating system will be trusted to support such processing for a number of reasons including: 1) access to the Octopus Network and its peripherals is strictly limited to personnel who can gain access to a physical security area, 2) Octopus Network/ LINCS software is written in-house by cleared personnel, 3) no outside access to the Octopus Network via public telecommunications facilities (i.e., the public telephone network) is allowed, and 4) various other security measures.

LINCS will also be layered onto the operating systems of vulnerable nodes. In a number of cases these vulnerable nodes are interconnected to form compartmentalized vulnerable LANs. Physical access to vulnerable nodes and LANs is limited to personnel with the appropriate security clearance and need-to-know requirements.

LINCS has been designed to support the three principles of distributed system security given above. Gateways with trusted hardware/software or trusted end-to-end encryption equipment attached to vulnerable gateways will be used to guarantee packet origin addresses up to an appropriate level of granularity and packet security levels, trusted gateways and routing nodes will enforce constraints on the paths packets take through the communication subsystem. The LINCS software that must be closely scrutinized in trusted nodes is limited to kernel code implementing the interprocess communication service, capability protection mechanism (see below), peripheral equipment interface and primitive kernel scheduler.

Since access to objects such as files, directories and processes is controlled by their servers, the code that implements servers must also be correct. Note, however, that errors in server code only affect objects managed by that code. Thus, the object-orientation of LINCS also provides a fail-soft security property.

5. Discretionary Security in LINCS

Discretionary security in LINCS is founded on capability-based resource access control. As indicated above, whenever a customer process desires to access an object (i.e., a resource), it must present a capability for that object to the object's server. Capabilities must be protected so that they are subject to neither forgery nor useful theft [WAFL79, DOFL8Ø, NESD82].

LINCS capabilities can be passed between any two processes that are able to communicate. Note that communication between processes on vulnerable distributed system parts in different compartments is not possible because of the constraints imposed by end-to-end encryption. Thus, capabilities cannot be passed between processes running on vulnerable nodes in different compartments. The futility of attempting to prevent the passing of capabilities or any other form of access rights between processes that can communicate is discussed in [DONJ81].

Capability-based access control has the advantage of supporting the principle of least privilege. That is, when a capability is passed, the receiving process is granted access to exactly one resource. A capability can also be passed to a server along with a request to issue a new capability with reduced access rights. This allows even finer control in access rights management.

Since LINCS can be layered on top of vulnerable nodes as well as being supported directly on nodes running trusted software, capabilities cannot be protected by simply storing them in node kernel space (i.e., all node kernels may not be trusted) [WAFL79]. Therefore, LINCS capabilities are stored in process space. Since capabilities are protected against forgery and useful theft by the capability protection mechanism, storing them in process space introduces no security risk. Doing this has the further advantage of considerably simplifying the LINCS service protocols.

6. A Proposed Distributed System Security Policy Model

The discretionary and non-discretionary security mechanisms of LINCS have been integrated into a proposed distributed system security policy model [FLEJ82]. While it is beyond the scope of this paper to describe this model in detail, a number of its attributes will be mentioned.

Processes running on trusted nodes are assigned both a primary and secondary security level (primary level is greater than or equal to secondary level). A process is prevented by the node kernel from either sending or receiving packets labeled with a security level higher than its primary level and from sending packets labeled with a security level lower than its secondary level. The effect of this policy is to give a fine gradation of administrative control over exactly which processes can communicate.

Communications between two vulnerable parts cannot make use of primary and secondary levels since there is no trusted code available that can enforce the necessary security level discrimination. This does not introduce a security problem, since physical access controls on interconnected vulnerable parts insures their security.

The model also contains rules to insure packets do not travel over channels or through nodes that are not rated to handle information at the packet's security-level. Other rules to control user logon and server/customer interaction as well as specify acceptable network topologies are included in the model. Readers interested in a more detailed description of the model should consult the cited reference.

7. Summary

LINCS is an object-oriented distributed operating system specifically designed to support the multiplexing of multi-level, multi-compartment computation over geographically disperse, heterogeneous and multiply-administered computing equipment. It supports fundamental principles of distributed system security and provides both non-discretionary and discretionary security mechanisms.

Non-discretionary security in LINCS is based on an amalgamation of trusted and vulnerable nodes and LANs protected by a mixture of trusted hardware/software and end-to-end encryption. Discretionary security is founded on capability-based resource access control.

8. Acknowledgments

Throughout its development all of those involved in the design of LINCS strove to insure that it could multiplex multi-level, multi-compartment distributed applications. This led to a stimulating environment in which many problems in distributed system security were examined and discussed. It was in this environment that the concepts and principles expressed in this paper were developed and codified.

9. References

[DOFL8Ø] Donnelley, J. E., and Fletcher, J. G., "Resource Access Control in a Network Operating System," Proc. of ACM Pacific '8Ø, Moon-Lith Press, Mountain View, CA, 198Ø, pp. 115-125. (also, LLNL UCRL-84319, Rev. 1)

[DONJ81] Donnelley, J. E., "Managing Domains in a Network Operating System," Proc. of the International Conference on Local Networks and Distributed Office Systems, London, 1981. (also, LLNL UCRL-85562)

[FLEJ82] Fletcher, J. G., "A Security Policy for Distributed Systems," Lawrence Livermore National Laboratory UCID-19544, April, 1982.

[FLWA82] Fletcher, J. G. and Watson, R. W., "An Overview of LINCS Architecture," Lawrence Livermore National Laboratory, UCID-19294, Nov. 15, 1982.

[HHDU82] Heiden, H. B., and Duffield, H. E., "Defense Data Network," Proc. of Eascon 82, Washington, D.C., 1982, pp. 61-75.

[KENS8Ø] Kent, S., "Security in Computer Networks," in *Protocols and Techniques for Data Communication Networks*, Englewood Cliffs, N.J., Prentice Hall, 198Ø, pp. 369-432.

[NESD82] Nessett, D. M., "Identifier Protection in a Distributed Operating System," Proc. National Electronics Conf., Chicago, Oct., 1981, Vol. 35, also appeared in Operating Systems Review, Jan., 1982.

[NESD83] Nessett, D. M., "A Systematic Methodology for Analyzing Security Threats to Interprocess Communication in a Distributed System," IEEE Trans. on Communications, Vol. COM-31, No. 9, Sept., 1983, pp. 1Ø55-1Ø63.

[WAFL79] Watson, R. W., and Fletcher, J. G., "An Architecture for Support of Network Operating System Services," *Computer Networks*, Vol. 4, No. 1, February, 198Ø, pp. 33-49.

[WATR78] Watson, R. W., "The LLL Octopus Network: Some Lessons and Future Directions,"Proc. 3rd USA-Japan Computer Conference, San Francisco, CA, Oct. 1Ø-12, 1978. (also, LLNL UCRL-81Ø67).

[WATR8Ø] Watson, R. W., "Distributed System Architecture Model," Chapter 2 in *Distributed Systems - Architecture and Implementation*, Springer-Verlag, New York, 1981.

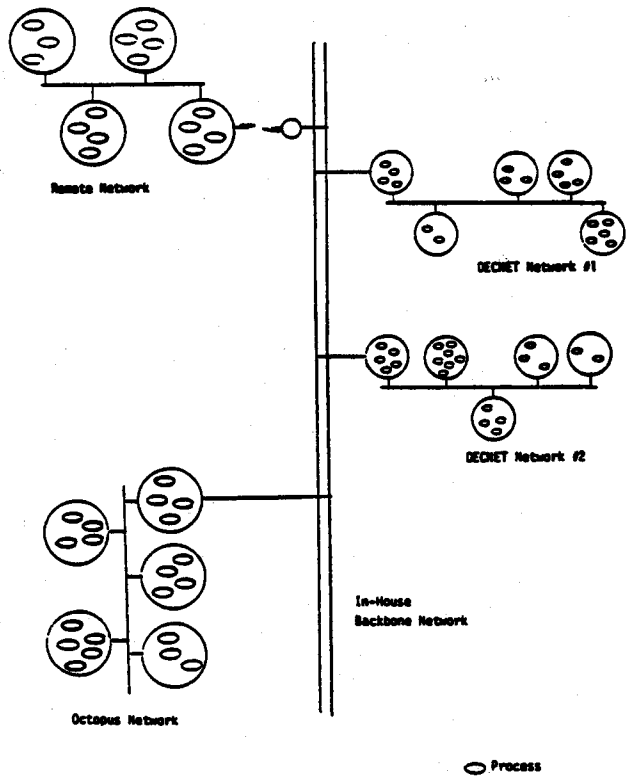


FIG. 1. PHYSICAL CONFIGURATION

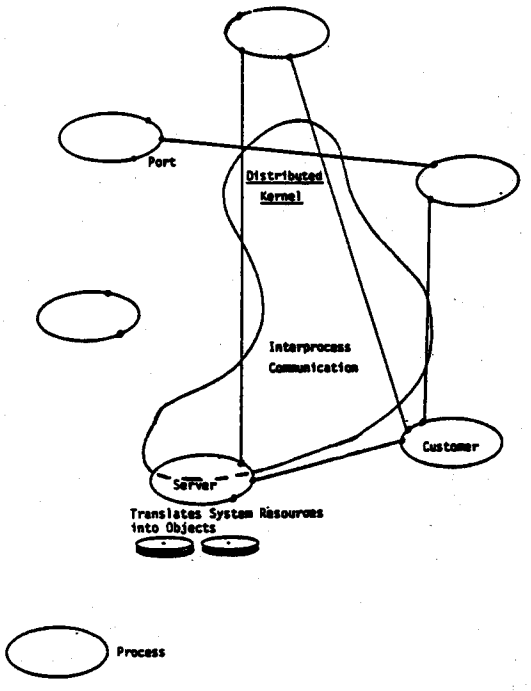


FIG. 2. LINCS LOGICAL CONFIGURATION

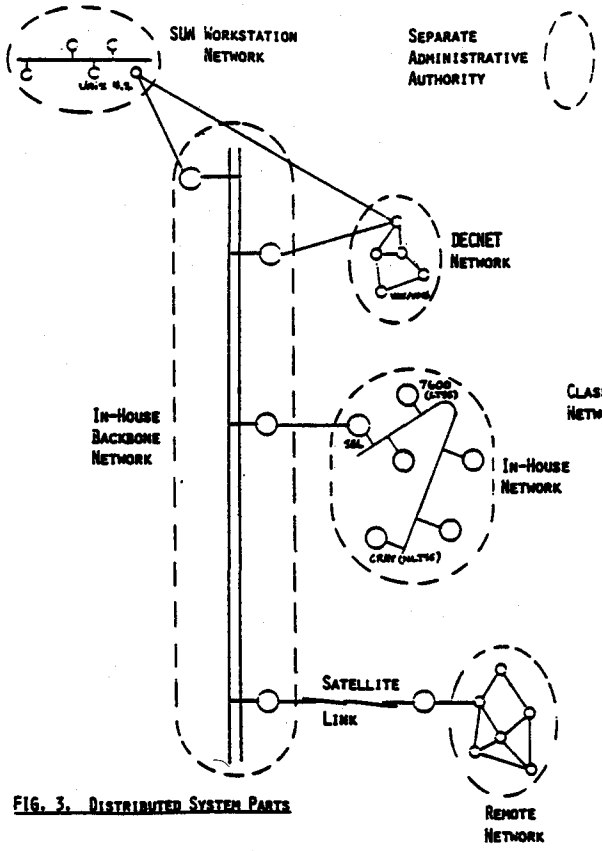


FIG. 3. DISTRIBUTED SYSTEM PARTS

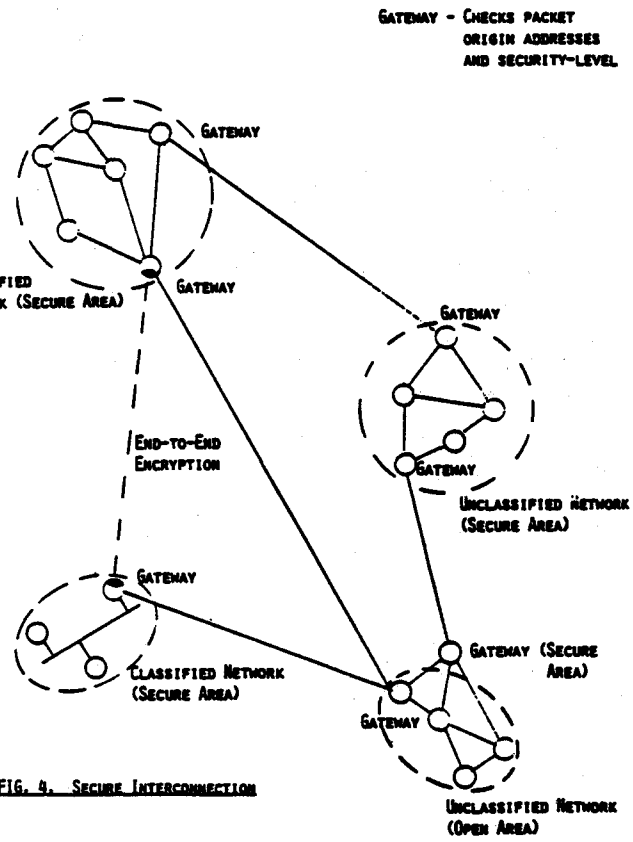


FIG. 4. SECURE INTERCONNECTION

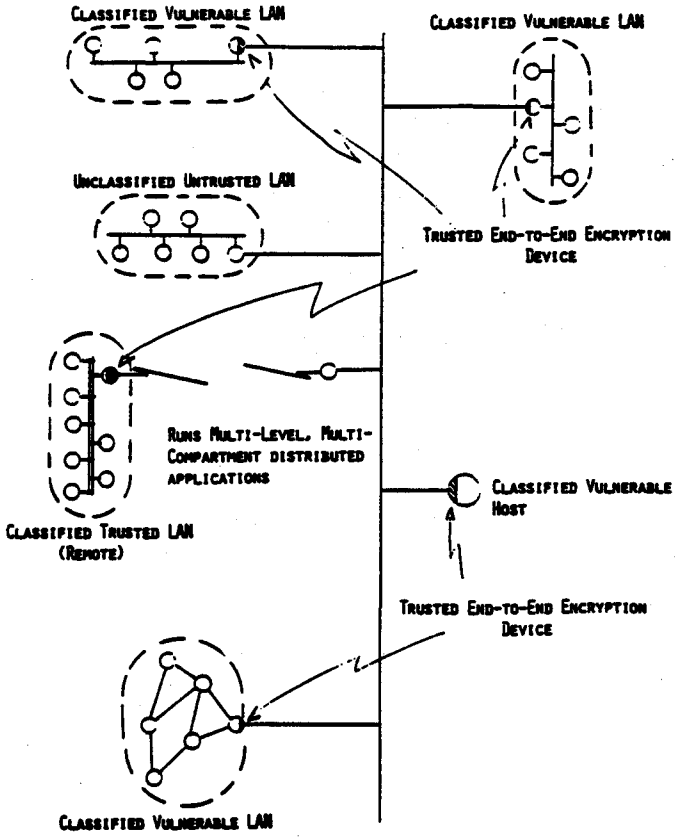


FIG. 5. INTERCONNECTING VULNERABLE PARTS

VERIFICATION TECHNOLOGY TRANSFER

Richard Kemmerer

University of California, Santa Barbara

When Mary Schaefer asked me to participate in a panel on verification technology transfer my first reaction was that we have come a long way. That is, it seems like only yesterday that the transfer of verification technology meant that some of the key personnel from SRI were now working at the University of Texas or vice versa.

Seriously, the transfer of the formal specification and verification technology from the research laboratories to the marketplace is occurring today. To aid in this progress the DoD Computer Security Center has made a commitment to support formal specification and verification tools. The center is going to make available "stabilized" and supported versions of FDM, Gypsy, and HDM hosted on Multics and accessible over the Arpanet. Multics was chosen as the development and verification environment for the Center because of the inherent security requirements.

I would like to stress that these tools are not end products. However, they are more stable than lab products, and have been used by other than the developers. In particular, Ford Aerospace, Honeywell, RCA, Texas Instruments, and TRW have used or are currently using these tools.

At this time the schedule for the availability of the tools on the Multics system are as follows. The HDM tools will be available by mid '84, the FDM tools by October '84, and the Gypsy tools by March '85. Each of the panel members will discuss how realistic these dates are. I would like to remind you that these dates are for the Multics version of the tools and that all of the tools are currently available on other systems through either Arpanet or Tymnet. Any serious users that can not wait for the Multics versions should contact the center to find out how to get access to the current versions.

The other members of the panel are:

Debbie Cooper - from System Development Corporation who will discuss FDM and the Ina Jo language,

Don Good - from the University of Texas at Austin who will discuss Gypsy; and

Peter Neumann - from SRI International who will discuss HDM and the SPECIAL language.

The panelists have been explicitly asked not to discuss the details of the tools, but rather to discuss the risks involved in making the tools available on Multics, what kind of people are best suited for writing specifications and performing proofs, what kind of training is necessary, and what kind of classes and documentation are available. Those of you who are interested in an overview of each of the specification and verification tools should consult the article entitled "Verifying Security" by Cheheyl, Gasser, Huff, and Millen that appeared in the September 1981 issue of *Computing Surveys.*

The format that we have chosen for this panel session is to have each panelist talk for approximately ten minutes. After that we would like to open it up to questions from the floor.

THE FORMAL DEVELOPMENT METHODOLOGY

Debbie Cooper

System Development Corporation

The topic of my discussion is the conversion of our verification tools to the Multics system. However, since many of you may not be familiar with SDC's verification methodology and tools, I will give a very brief overview of these before addressing our plans for the Multics conversion.

The Formal Development Methodology (FDM) is an integrated methodology for the design, specification, development, and verification of trusted software. FDM is supported by several tools. Top and lower level design specifications are written in the Ina Jo Specification Language, a non-procedural assertion language based on first-order logic. Ina Jo uses a state machine model, in which the system functions are represented as state transition functions. Each Ina Jo specification includes a representation of the entire system design, at different levels of detail. A top level Ina Jo specification is highly abstract, with minimum functional detail. Each lower level specification introduces greater detail than the previous level. The number of lower level specifications may vary, depending in part on the complexity of the system, and ultimately, on the personal judgement of the specification writer. The lowest level (most detailed) Ina Jo specification is followed by the Implementation Specification, which essentially provides the correspondence between the Ina Jo abstract design specifications and the higher-order language (HOL) code.

FDM differs from some of the other formal verification technologies in that the correctness requirements for the system are not built into the tools. Instead, the system correctness requirements are supplied by the user as part of the Ina Jo specifications. The critical requirements, such as the security policy model, are expressed as Ina Jo "criteria" and "constraints," and are included in the top level specification only. Criteria are state invariants, conditions or properties that must be true for every state of the machine. Constraints stipulate relationships between consecutive states. (Constraints were added to the language as a direct result of early applications of the tools.) Supplementary requirement assertions may be included in any level of specification, expressed as Ina Jo "invariants."

Lower level specifications may contain any of the language elements of the top level specification with the exception of criteria and constraints. In addition, each lower level Ina Jo specification contains a set of "mappings" which relate elements of the lower level specification to elements of the next-higher level of specification. Some of these mappings are generated by the Ina Jo Processor, however most are supplied by the user and are included in the lower level specification.

The Implementation specification is the final stage in the formal specification process, and provides the correspondence between the abstract design specifications and the implementation program. Implementation specifications are written in Inamod, a modification of the Ina Jo language for HOL code, and mappings between elements of the lowest level Ina Jo specification and the HOL program.

All specifications are processed by the Ina Jo Specification Processor, which generates both error messages for users and candidate theorems which are proved with the assistance of the Interactive Theorem Prover (ITP). These candidate theorems are of two varieties. "Consistency" theorems are designed to demonstrate that the various elements of each level of specification, and the mappings between levels, are not contradictory. "Correctness" theorems assert that each state transition preserves the stated requirements, and that lower level mapped state transitions are correct implementations of their higher level abstractions. Given the lowest level Ina Jo specification and the Implementation specification, the Ina Jo Specification Processor generates entry and exit assertions for each HOL procedure. These entry and exit assertions and the HOL code are then input to a Verification Condition Generator (VCG), which produces the verification conditions, theorems which assertthat each HOL procedure satisfies its exit assertions assuming its entry assertions are true upon invocation of the procedure. A different VCG is needed for each HOL. We have recently implemented our first VCG, for York Modula, and are in the process of testing it.

Each level of specification is fully verified before beginning preparation of the next lower level of specification. All candidate theorems, including the verification condition theorems, are verified (using mathematical proofs) with the assistance of the Interactive Theorem Prover (ITP).

The process of writing and verifying Ina Jo specifications tends to be highly iterative. Trying to verify the specifications occasionally reveals bugs or flaws in the design, or in the specification of the design, which require modifications to one or more levels of specification and re-verification of the modified specifications. This approach, however, has two very important advantages. Abstract specifications are an information or detail-hiding mechanism, and thus make it possible to formulate a concise overview of the design of large and complex systems. In addition, it is far more cost effective to identify and fix design flaws early in the design phase than it is to correct these effors in the code, or to debug the system after it has been implemented.

FDM has been used for the verification of several trusted systems. Past applications include top and second level Ina Jo specifications and proofs for a color change controller, a secure packet switch network, OS kernels, and multilevel secure communication systems.

The Ina Jo processor and the ITP are expected to be ready for use on Multics by 1 October 84. Our tools are written in CWIC, an SDC compiler generation system presently compatible only with IBM software and hardware. Prior to the Multics contract, we were in the process of transporting the FDM tools to new systems. The

conversion path we chose was to build a translator for converting the CWIC tools to the C language. We chose C as our target language because of its versatility and portability, and the Digital Equipment Corporation VAX as the implementation machine. The translator has been built and implemented, and we are currently doing checkout testing of the tools on a VAX 78Ø running under UNIX.

For the Multics conversion, we considered several possible paths, ranging from porting the Bell C compiler to Multics to rewriting the tools in PL/1. The conversion path we chose was to modify CWIC to generate PL/1. There were several key factors in this decision: PL/1 is the most commonly used and most efficient higher order language available for Multics, and should generate more efficient code on Multics than any C compiler that we could expect to have available. In addition, an initial belief that writing a C compiler for Multics would be significantly easier than the CWIC-PL/1 approach proved fallacious. Both efforts appear to require roughly the same number of lines of code. In addition, by using the CWIC-PL/1 translator approach, we are able to maintain a common source for our FDM tools. Hence, we can assure all FDM users the same functionality regardless of the computer on which the tools are implemented. Periodic updates to the source, which reflect both error corrections and enhancements, as well as check-out testing, will be conducted at SDC.

The greatest anticipated risk involved in the Multics conversion is performance. From a user's perspective, performance degradation is a more serious problem for users of the ITP than for users of the Ina Jo processor. The composition of mappings tends to cause an explosive increase in the size of theorems for lower level specifications, and the proof of these theorems can be a tedious and lengthy process. Multics is on a much bigger machine than those on which the tools have been running, which may provide some remedy. In addition, we have plans to improve the speed and efficiency of our theorem prover.

Most users of the FDM tools have been SDC employees. With the Multics conversion, the tools will be available to a wider community of users, which raises the issue of how to choose and train potential FDM users. We have had very little difficulty in training people to use the ITP, and to learn the Ina Jo syntax. The most difficult aspect of using the tools is the abstraction process: recognizing the difference between detail-hiding and loss of rigor, determining how much detail is sufficient for each level of specification, isolating the critical requirements of a system from the myriad of (sometimes conflicting) functional requirements requested by the customer. The most important factor that surfaces from our experience in choosing and training specification writers is, "Know the system!" SDC has training courses and documentation on how to use the tools. Any inquiries should be directed to the DoD Computer Security Center.

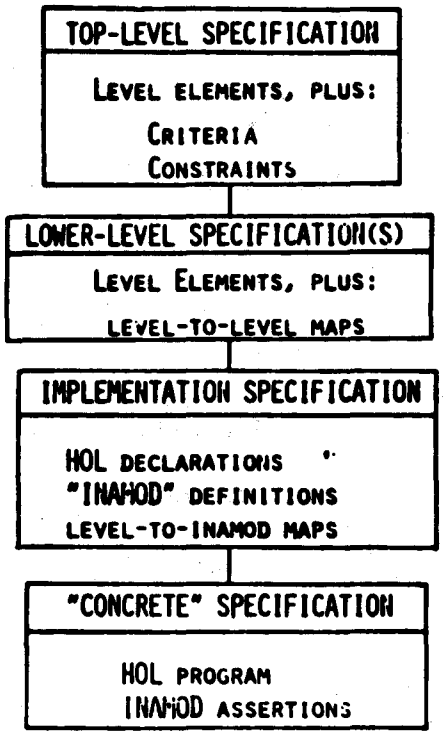
FDM

FORMAL DEVELOPMENT METHODOLOGY

TOOLS

INA JO SPECIFICATION LANGUAGE
INAMOD -- INA JO MODIFIED FOR HOL
INA JO LANGUAGE PROCESSOR
INTERACTIVE THEOREM PROVER (ITP)
VERIFICATION CONDITION GENERATORS (VCG)

SPECIFICATION STRUCTURE



Slide 1

FDM Applications Include:

- Color Change Controller
- Secure Packet Switch Net
- OS Kernels
- MLS Communications Systems

MULTICS Conversion Issues:

- Conversion Path -- CWIC-to-PL/1
- Performance -- ?

- FDM Users
- Training
- Documentation

Slide 2

THE HIERARCHICAL DEVELOPMENT METHODOLOGY

Peter Neumann

SRI International

The thing to realize here is that we are living somewhere between research and development. In fact, we are trying to live in both worlds at the same time. I stress the middle word in the expression "Research *and* Development." It is, in fact, the "and" that is most important here.

The HDM effort arose originally out of provably secure operating systems starting in 1973, and the great emphasis at that time was on the design of the secure system. A lot of the methodology arose pretty much as a by-product of that, and as a result there were never really extensive research funds or even development funds for making sure that the methodology and the tools were suitable for development efforts - for example, in terms of having extensive documentation and careful examples that would demonstrate the utility of the methodology.

Here we are ten years later. I'm assuming that all the people in the audience know nothing about HDM and all of the empty chairs are due to the fact that all of the people who are not in here are out in the lobby talking and know everything about HDM. But, I will still stick to the guidelines set up by Marv Schaefer and Dick Kemmerer, which are not to talk about the methodology in any great detail - despite the fact that there may be many of you who don't know much about it. Let me just say that HDM is a formal methodology that encompasses formal requirements, formal specifications, and formal proof. It includes proving consistency of specifications with formal requirements - not just the MLS that you have come to know and love, but other policies that may, in fact, be represented in the scope of the methodology (although the tools that exist today do not support other policies very well). Also, HDM is intended to work with various programming languages. And again, the tools that exist today do not support the development of code proofs in a truly compatible way with the existence of design proofs. You can do design proofs and you can do code proofs. There is a problem of doing them in a compatible path through a consistent, coherent set of tools. Recognizing this, the Center has finally decided they would like a coherent set of tools. In fact, at the moment they are funding an exercise to come up with something we're calling Revised SPECIAL, which is an up-grading of the specification language SPECIAL, which is a part of HDM. Let me make a careful distinction between HDM (which is the methodology which exists as a set of guidelines using types and abstraction and hierarchical structure and good things like that based on Parnas, Wirth and Hoare, and others) and SPECIAL, which is the specification language.

Now, SPECIAL has, in fact, been used along with HDM in several systems. Just to give you some indication of the fact that this is not totally a research effort, we have the SCOMP effort in which the top-level spec of the kernel has been specified and proven using SPECIAL, although the trusted processes are going to be done in GYPSY. The KSOS kernel design proofs were, in fact, done by Ford and SRI. The SACDIN kernel proof was done by Sytek. PSOS (Provably Secure Operating System) was specified only, but became the vehicle under which HDM and SPECIAL were developed. Contrary to a comment made earlier, there are few real-time secure systems. (For example, SRI did a design, called TACEXEC, for the Army at Ft. Monmouth. I have also added a couple of other systems to the list, one that is known affectionately to the people in the military computer family business (MCF) as MCFOS (pronounced McFos). There are, in fact, two competitive efforts: one, RCA, with Richard Platek and the Odyssey people; and the other, TRW. Both of them appear to be using SPECIAL.

So, the tools do exist. They are reasonable to use at the moment, if you are interested in proving a property that is effectively the MLS property that you've been told so much about. That leaves us with various questions about how to use them and when to use them and when they're appropriate and what the risks are. Dick showed a slide that said that there would be a version of the existing tools running on MULTICS. We have stated that we will have that ready nine months after we have a contract. At the moment, we have no contract, so the July date would have been okay if something had started in October.

As I said, we are pursuing a somewhat different direction in augmenting the specification language and adding to it new features such as parameterized modules, and various other features that will make the specification language much more appropriate for dealing with distributed systems and concurrent systems. Marv explicitly asked me not to mention an example of code proof applied to that kind of an effort (SIFT), and I'm not going to mention that. But I do want to point out that there's still a lot of research to be done in this area in terms of dealing with models of security that are more close to reality, when you're dealing with multi-level secure databases, with aggregation, inference, and down-grading, the strict MLS policy as it exists is not capable of handling the desired policies. So, our new generation of tools that will support the Revised SPECIAL language will, in fact, be able to address some of that. Those tools will run on the DEC Twenties as the existing tools do. They will also run on MULTICS with an IBM PC as an optional front end, and also on a Symbolics machine. So, we are trying to get the tools more widely available, both the old tools and the new tools.

We find one of the risks in using the old tools is that there is no strict, formal basis for the MLS tool itself. One of the things we're hoping to do for the second generation is to have a formal basis for the tool. In other words, a formal representation of the security policy which is then fed to a table-driven tool that generates itself as essentially a meta-tool, so that the tool itself has a more rigorous mathematical basis. A question raised this morning was the old question of who "Schaefes" the Schaefer? And that was, what do you do about verifying the verifier? The answer could be given that one could build a proof checker that would check through a lot of the design proofs and the

code proofs, a rather simpler beast than the verifier itself. But it still leaves a lot of potential problems. Does the model really represent the policy that you're trying to deal with? Do the people who are trying to develop the system understand the policy? Well, that should be smoked out by the design proof tools, of course, if they've misrepresented it. But, in general, the notion of what might be called cumulative confidence, as George referred to in slightly different terms this morning, is one where there is always some risk in this process. At the moment, the proof procedures are good at deleting storage channels, but not very good at smoking out Trojan horses, trap doors and things like that, and we believe that some of the new directions will make that much easier. At the moment the tools are somewhat difficult to use. The documentation is not as copious as it might be. I recommend that you ask some of the people in the audience who have suffered with trying to use the tools as to what the pitfalls are. There seem to be quite a few pitfalls. On the other hand, we believe that the tool, if used properly for multi-level security, is sound; but we will be much happier when we have one that is more formally based.

As far as ease of use goes, we also believe that the coming tools will be easier to use because of their greater attention to the notion of their use as a part of the design, whereas the first specification language really grew randomly: everytime we needed another ability in order to specify some curious aspect of PSOS, we simply stuffed it into the specification language. So, the language is not as beautifully structured as you might like it to be.

So with all of those caveats, we believe that the tools are useful and we believe that the fact that they have been reasonably widely used does bear on future efforts in this direction. Let me stress again in closing that it's the combination of research and development that is important. John Lane mentioned a symbiosis among all his "ilities." In fact, most of his "ilities" were antagonistic, so they're not really symbiotic; but the approach here is one where, in fact, the research and the development should be in symbiotic relationship with each other.

SRI's Hierarchical Development Methodology (HDM)

- Unified approach to design, implementation, and verification
- Formal requirements (model)
- Formal specifications (design)
- Early design evaluation
- Support for real systems needs
- Various programming languages
- Tools enforcing the methodology

Slide 1

SRI Historical Perspective

HDM SPECIAL MLS — BOYER-MOORE — STP SIFT PROOFS

Revised SPECIAL

Slide 2

Specification and Assertion Language: SPECIAL

- Early specification language
- Based on Parnas, Hoare, Wirth; hierarchically structured abstract data types (modules)
- Tools INTERLISP-based, useful, run on TOPS-20/TENEX; Multics contemplated
- Spec proof and code proof components not well articulated
- No formal semantics

Slide 3

Tools Developed at SRI (1975-1982)

- SPECIAL Specification checker
- HDM Interface checker
- HDM Hierarchy checker
- HDM Representation checker
- Formula generator for proving SPECIAL specs satisfy MLS
- Boyer-Moore Theorem Prover
- Verification condition generator (internal form for LISP, Modula)
- Fortran VCG
- Fortran verification system
- Pascal verification system
- Meta-VCG, used for Pascal verification system and Ada program visualization
- STP Theorem Prover

Slide 4

Current Status

HDM used in various secure systems (SCOMP, KSOS, SACDIN, PSOS)

Old SPECIAL tools useful, notably spec checker, MLS SPECIAL design prover Pascal code prover incompatible

Revised SPECIAL defined; formal syntax, semantics exist; tools under development; experimental system expected in mid-1984

Slide 5

Status of HDM Use

SCOMP [KSOS-6] (Honeywell), Kernel design proofs. [Trusted process proofs will use GYPSY]

KSOS-11 (Ford), Kernel design proofs (SRI/Ford) Kernel used in test sites. Performance upgraded (Logicon)

SACDIN (ITT/IBM), Kernel design proofs (Sytek)

PSOS (SRI), Specified. Basis for Honeywell's Secure Ada Testbed

TACEXEC (SRI), Secure real-time system, design only (Army ECOM)

Slide 6

Status of STP Use (Experimental)

SIFT (SRI, Bendix, NASA), hardware and software running at NASA; extensive design proofs over various levels of abstract models down to SPECIAL specs, plus prototype code proofs carried out for a simplified version of the running system

Being superseded by Revised SPECIAL environment

Slide 7

Revised SPECIAL

Very simple conceptually

Strongly typed

Parameterized theories

Extendable

Specifications more succinct

Formal syntax and semantics provide formal basis for unified tools

Good documentation required

Slide 8

Revised SPECIAL Potentials

Parameterized modules permit
2nd-order logic
Monadic set theory
Sequences, bags, tuples
Program variables
Operations with side-effects
Pascal and Hoare sentences
Temporal and interval logics

Language well-suited for use with advanced security policies for trusted computing bases - e.g., using Goguen-Meseguer policy formulations

Slide 9

New Verification Environment

Tools MACLISP-based; will run initially on Multics (with IBM PC as optional front-end), Symbolics, TOPS-2Ø, TENEX Later on VAX,...

Multiwindow mouse-oriented interface, usable on conventional terminals

Structured editor

Database with version, dependency, and configuration control
High-level expression and debugging of proofs

Unified paradigm for design and code proofs

Slide 1Ø

Advantages of New Environment

Unified environment for specification and verification

Required expertise better matched to available people

Better man-machine symbiosis - user-friendly environment

User provides understanding of problem and proof

System provides rigor, record of what has been done, complete decision procedures for useful theories

Slide 11

PANEL SESSION - SECURITY TECHNOLOGY IN THE MARKETPLACE

Moderator - Dennis Steinauer

Computer Specialist, Computer Security Management and Evaluation Group

Institute for Computer Science and Technology, NBS

Panel Members:

Peter Browne - EDP Audit Controls, Inc.

Edward Zeitler - Security Pacific Bank

Randy Sanovic - Mobil Corporation

F. Lynn McNulty - U.S. Department of State

INTRODUCTION

The primary objective of this panel is to try to verify those areas in the market for security products and services which really are weak. We want to look for holes in the market, as opposed to only discussing the products that are already available. However, I would define as one of the holes in the market those areas where there are products that people aren' t using for one reason or another. But, equally important, perhaps more important, is identifying products that aren' t there that we think should be there. What I' d like to do first is make a couple of assertions, and then I' ll introduce the panel and let them discuss the marketplace briefly from their individual perspectives.

Those assertions are, first, that security protection for automated information systems can be improved significantly through the use of commercially available hardware, software, or related products and services. Probably a fairly safe assertion for all except one person in the audience, I think.

The second assertion is that the use of commercial products in general will provide a more cost-effective and efficient level of protection than controls that are developed by user organizations, especially for the growing population of small, non-technically oriented users.

The third assertion is that there are some significant commonalities in the needs of both military and non-military, both government and civilian - factors which can be exploited or can be well served by commercial security products and services.

So, those are the main points around which we will base our discussion. We don' t have individual topics for each speaker, and this was done intentionally. I think it' s important to have a general discussion of these topics.

The purpose in selecting these panelists was to talk about what the non-military community is looking for in terms of the security products market. Each of them will discuss very briefly their views of what' s happening in this market and what its problems are.

Peter Browne

1) I see two differences between the requirements of the commercial sector and of the DoD:

a) One is the drivers, the motivating mechanisms, the various miltary and intelligence regulations in DoD, and in the commercial sector, the drivers are much more informal, and unfortunately, based on risk perception - the driving forces are much more financially based.

b) Secondly, the mechanisms in the commercial sector are probably not as mature as they are in the DoD environment. There's not as much formal specification.

2) There are six axioms that must be dealt with or you really don't have a very complete protection capability: policy, marketing, identification, accountability, assurance, and continuous protection.

3) The control objectives relating to accountability and assurance are exactly the same mirrored across any sector.

4) Assurance at class C2 is really what the commercial sector requires.

5) We need controlled access between named users and named objects, and granularity at least at that level, and all of the caveats and mechanisms that this requires.

Ed Zeitler

1) It's very hard to explain to management that we, a bank, need high-tech solutions, since we do not have high-tech crime. Solutions that affect our customer base or those that are expensive are hard to justify, and we have to remain competitive.

2) Jim Anderson mentioned yesterday that the CEO must be aware that the Criteria is derived from management-controlled principles. However, getting the line encrypted,

getting the access control power fees actually enforced, especially in the small data center like the 414 people who are off running around the contryside: that's middle management, and that's where we really have to cancel the problem.

Randy Sanovic

1) Some of the needs that the military has, and some of the things they're working on, are the same as private industry's needs. You can argue about the qualitative or the quantitative differences, but I'm not going to argue about that, because certainly there are pluses and minuses there. When we take a look at what the government is doing, we ask ourselves questions like, "Should we be interested in doing some of that?" There is a definite need to work together on these things.

2) Another area we're very interested in is integrated security products that can protect the user or executive workstation accessing multiple databases across multiple DP environments and domains, and products that deal with data security and telecommunications security.

3) On the larger vendor equipment level, we're not only interested in data security at the IBM level or the Honeywell level, but at the CDC level, the Cyber level for system processing, the distributive mini- and micro-base level. We're not distributing main frames for people to use, we're distributing mini's and micro's.

4) If we want security products, we want to be able to monitor what goes on in the systems. We're concerned about technical correctness and simplicity.

5) We have to be educated that a demand exists.

6) Companies still need to recognize the value of the data they process, not only the data itself, as Peter mentions, but the money they're moving around.

7) The products that can meet industry's requirements must fulfill three needs : a) they must be technically secure, b) they have to be tremendously cost-effective, and it has to be a full range of equipment: wide band, narrow band, high speed, low speed, data, text, voice, and facsimile, not a plethora of products that have been piece-mealed into your operation, c) products that can solve point-to-point and end-to-end encryption designs and network needs in that area - that combine data texts and voice facsimiles, and a secure technology that doesn't have to be administered through crypto centers with all the loss of the related manpower sources.

8) To a large extent companies still have to learn to use the products that the hardware and software vendors have supplied them, and then go on and work with products that may be coming from the military and federal marketplaces.

9) I think something needs to be done to drag these products out of the government sector. For instance, some of the evaluations that are being done on PCs, security passwords, audit trails, etc., will help direct proper technology to the commercial marketplace.

F. Lynn McNulty

1) We at the State Department are very heavily involved in office automation and the distributive processing environment. We are placing a lot of equipment out in our overseas embassies, and we're sharing this equipment with Commerce, Agriculture, Defense, as well as State Department personnel.

2) What we're looking for is really some way of getting a better handle on the office automation environment.

3) The most sensitive data, and I'm looking at sensitivity from the point of view of unauthorized disclosure, is sitting on the office system, the word processing system, or the system that is used heavily for word processing. It may not be on the big main-frame computer or the mini-computer department.

4) The kinds of products we're looking for are ones that have the ability to control who gets on the system. There's no reason why, when somebody comes in on a Saturday afternoon to work on a system, all the terminals come up at the same time.

5) We're also looking for some ability to control access to all of the libraries that are present on the word processing system; that's where some of the most sensitive data in anybody's organization is.

6) We've successfully addressed the Tempest problem in the office environment, but at the same time, we haven't made the translation that says, if you're going to give us systems that are approved for processing classified data, then let's clean up some of the software problems and give us a system that provides need-to-know enforcement as well. I think that's equally important in the commercial environment as it is in the government or national security arena.

7) We're also looking for the improved capability to do backups. There are some products coming down the road in this area to improve the speed and the ability to backup, but we would definitely like to see some sort of capability to improve the backup for these systems as well - a simple audit trail capability.

8) We're also looking for materials in security education addressed to the office environment. There is very little out there in the way of films, publications, posters, and things of this nature that we can use to orient people who are users of office systems.

9) We'd also like to see something in the way of security management guides for the more sophisticated office systems; a book that explains what the system gives you in the way of security and how to implement it.

1Ø) Something that will give some kind of security audit capability, some ability to go in and do on-site security evaluations is also needed.

QUESTION AND ANSWER SESSION

Dennis Steinauer: Bruce, you being from the vendor community, what is it that your organization sees as a need? Either what is being demanded, or what you think someone ought to be demanding?

Bruce Warner: Our view is much narrower and more precise. We're focusing on essentially one product, the IBM PC, which is an outgrowth of a concern about all of the sensitive data on a manager's desk or a workstation that isn't protected at all with floppies laying around as mentioned before. In order for such a system to work it should be reasonably easy to use, otherwise it won't be used. It should be relatively transparent to the user. That's the essence of a product we call PC Lock and that's our focus and our concern about security.

Dennis: Going back to what you had said, Lynn, about getting into small systems, particularly personal computers, executive workstations, that sort of thing, there is obviously a problem of physical control and it becomes very difficult if not impossible to guarantee the integrity of any software that may be implementing access controls. How important do you think that is? Some people would argue that it's senseless to try to put any type of access control, even using encryption, on or in a PC or small system environment simply because a technically oriented person could get in and disturb the integrity of the system and thereby circumvent whatever controls there are.

Lynn McNulty: Fundamentally, I guess I have to agree with that, but by the same token I think we have to realize that many of the users out there are not that technically inclined and we have to protect systems against them as well. In some cases we can supplement the system-oriented controls with physical security by making sure that we are able to control where the disk storage unit or the PCU happens to be located. Then what we're looking at is being able to control the activities and actions of users. I see no reason for saying that we can't do it, because there is always someone who will be able to technically defeat any measure we put in place. I think in a lot of cases, the people we're dealing with are not that sophisticated, but we still need to exercise some level of control over their ability to look at data that hapens to be resident on the system that they're given access to.

Randy Sanovic: I think I'd like to qualify that from a business perspective. Since we are getting more and more into downloading data off of our core applications out of multiple databases, people are reworking the same data and uploading it to the mainframe. So, obviously you've got a security problem there. It's not as if you're using a terminal strictly as a terminal. It's a smart terminal so it's critical.

Dennis: Any comments?

Ed Zeitler: From a banks's point of view, our threats are primarily internal threats. Our biggest threat is really disclosure and modification of data by people within our own organizations. We're not as concerned with wiretaps as we are with a teller who types in something and finds that it works kind of slick and pretty soon has walked out with $1Ø,ØØØ. So, when we're talking computer problems, we're talking internal almost always. Therefore, the control of these PCs, to download them, to take part of our data and actually distribute out to the PC is probably one of the most important aspects that we're trying to deal with.

Bill Smith (Cray Research): My question is directed to anyone on the panel. Mr. Brown spoke of commercial systems having to be at the C2 level to afford reasonable protection. My question is, to what extent does the private sector verify, in some analogous sense, security to the (quote) "C2 level" of their systems programmers and maintenence people, engineering and maintenance. I'm not talking about the tellers, I'm talking about the people who really have access to the system. It strikes me that there's no point to having computer security if you don't have people security.

Randy: Most of the companies that are doing anything with security have got an employee checklist, but in some countries we can only go so far in checking records. We try to go as far as we can when we go out and do vulnerability analysis with the use of our affiliates in 17 or 2Ø major computer centers. We take a look at their policies in this area within the legal privacy limitations of the country.

Ed: Speaking again for the bank. There had been a policy where you had security checks when you reached a certain level of management. It has just recently been changed so that sometimes clerks have to have a higher level of clearance than the vice-presidents. So, we are addressing those issues.

Dennis: One of the questions I had listed down here for possible discussion is whether or not there is technology that is now being used in the military that is applicable to the civilian sector but is not being used for one reason or another.

Ed: There are three guidelines, and I can speak for more than just the bank here: a product must be low-cost; it has to be user-friendly; and, it has to be fully supported by the vendor, preferably by the mainframe vendor. There aren't that many products out there today that meet those criteria. The operating systems are changing too quickly, applications are moving too fast. There are many products, I assume, that the military has available that we cannot really take advantage of, because they don't meet any of those three criteria.

Dennis: Any systems that are out there that aren't being used that, with some modification, could meet those criteria?

Peter Brown: Some of the testing and evaluation being done on network authentication systems have very definite applicability to the commercial sector. I haven't seen much of that in the commercial sector, but I think that once these get out of the laboratory into the marketplace, they will find that there will be a need.

Randy: I think the prime example of it is how it's moved out of the government, the commercial side of government, back to the public sector. Quite a few companies are interested in electro magnetic emanations and tempest work except that we don't understand how to use it.

Dennis: What do you use in that case, what product?

Randy: Well, largely because we can understand it and deal with it, it's usually hardware on dial-up, some lead, some dial-up line. And, until we can figure out how to administer end-to-end encryption and certain other things, we're probably not going to get into the resources required for that.

Peter: I think the encryption technology is going to find itself in the financial community very quickly. There was an ANCI standard passed for financial message authentication which, of course, uses encryption products. People are serious in the commercial marketplace about authentication to replace antiquated manual base touch key kinds of systems just to authenticate traffic that moves money.

Dennis: Do you think vendors think you're serious?

Peter: Yes, we know half a dozen vendors that are very serious. I've heard of products that are out in the marketplace already.

Ed: We're doing several tests right now with other banks and other agencies such as Fedwire. They're very serious about it, but quite honestly, I'll believe it when I see it. There has to be some standardization. We have to prove to our own management that what we're going to have will still be viable a year from now or two years from now. That's a very hard thing to sell right now.

Peter: I'm actually working with a bank that is implementing an authentication base system. We did some cost analyses and we came up with about 1. Ø197 cents per transaction. In that case, extremely cost-effective.

Dennis: If we had to list the specific characteristics that are really going to be needed, obviously there must not be enough products out there because people are continually asking for help. So, what characteristics are these products going to have to have so the people can buy them off the shelf?

Peter: I think it's been said: cheap, friendly, and transparent. But perhaps functionality has something to do with this, too. I think, given an investment in a processing environment or communications environment, you can't throw it all away. Perhaps you can put controls on and lock some doors by providing other functions, other attributes. For example, I know of an organization that has a problem with multiple access or multiple systems or subsystems in an IBM environment, all on the same machine - a control problem. The solution is almost deceptively simple. Build with what's already in the operating system, something they call the network deluxe software in which you log on to the network director and it provides an access mediating mechanism which will then do control tables, which is not necessarily the right way, but will provide access controls to the very subsystems depending on who the user is. And, all of a sudden, you're adding more controls and at the same time making it a lot easier to log off one subsystem onto another and not have to go through a whole interactive protocol everytime.

Lynn: I think one of the problems of working in the small system world and trying to develop products for that arena is the paper technology. By the time they've gone through the R&D and product development cycle, maybe the technology has changed. A system they originally developed that specific product for is no longer used or the vendor is pushing the next generation system. It's a moving marketplace and it's hard to develop products for the security arena when by the time you get it out for sale, people may not be buying and installing the product you developed.

Steve Glazeman (Ford Aerospace): This may be a somewhat naive question, however, it occurs to me that in discussing security products one of the major impediments to such purchases was the fact that there was no measure of return on the investment. Does anyone on the panel want to comment on what has transpired between the time I left the community and this particular point in time?

Randy: Last week in New York, we threw quantitative risk analysis out the door. I think that's the role return-on-investments strategy, risk analysis, has to play in the selection of whatever specific set of safeguards you throw around a given system or network. And, you have to relate it to what it is you're trying to protect, and what the value of the data is and ultimately the hardware assets. I think that's why the risk analysis concept or whatever methodology we use is somewhat crucial to assuring the appropriate allocation of resources.

Steve Glazeman: I would agree with that. However, I think I'm dealing with the other side of the issue. Just to say, once I have done a risk analysis, that is the best I can do at any particular moment in time, I need some sort of metric. Essentially, my risk analysis establishes my criteria against which I will measure whether the purchase product A developed product B or do something else.

Dennis: Well, frankly, I wonder if that basic issue isn't what people see as the fundamental difference in viewpoint between the National Security community and the civilian community. Particularly in the private sector there is a demand on the part of management for some sort of cost justification. What are you going to get for your money?

Randy: I guess there are three things I'd like to say. As of about eight years ago, many corporations started looking at how much they depended on computers and computerized data. And then some of them had to make a decision like we did, do you protect all of your data or some of your data? That has to start with the business people, the controllers, and the vice-presidents rather that the data processing people. If they make that decision, then perhaps you put an implicit data security system in. So, it's grown from data being the corporate asset and the company's depending on data processing to the protection of the data itself. And, it's kind of mushroomed or ballooned out from those directions, as far as I can see.

Dennis: That's one nice thing about having corporations that are almost as large as the government.

Peter: Despite what has been said, our risk analysis is not dead. There are tools and techniques and approaches to risk analysis or assessment which are going to overcome the deficiencies that we've seen in the past. I think every manager who makes a decision to spend three hundred or five hundred thousand dollars on contingency planning in a corporate environment has made a risk decision.

KEYNOTE SPEAKER Day 3
EMERGENCE AND EVALUATION OF SPECIFIC COMPUTER SECURITY PRODUCTS

Stephen T. Walker

President, Trusted Information Systems, Inc.

Mr. Walker is an electronic engineer and computer systems analyst with over 20 years experience in system design and program management. He is nationally recognized for his pioneering work on the Department of Defense Computer Security Initiative and for his extensive experience with the design and implementation of large scale computer networks and information systems. As Director of Information Systems for the Deputy Undersecretary of Defense for Communications Command, Control and Intelligence, he was the senior technical advisor for the Secretary of Defense for the World Wide Military Command and Control System (WWMCCS) Information System (WIS) and the Defense Communications System. In this position he established the technical and managerial structure for the modernization of the WIS, a five billion dollar, ten year effort and was responsible for a complete restructuring of the DoD data communications architecture, with the cancellation of the AUTODIN II program and the establishment of the Defense Data Network.

He is the founder and President of Trusted Information Systems Inc., a privately owned small business specializing in consulting on the development and management of information systems, computer networks, computer security, telecommunications, and related fields to the government and industry.

Mr. Walker is a member of the Defense Science Board Task Force on Defense Data Network, and of the Foreign Applied Sciences Assessment Center Panel on Computer Science.

Good morning. It's a real pleasure to be back here again. I remember getting together with Pete Tasker and his group at MITRE to plan the first few of these seminars. We wondered whether anything would ever come of our efforts. Then at the fourth seminar Admiral Inman announced the formation of the Center at NSA. That was exciting and I felt as if we had really made some progress. The fifth seminar was handled by NSA and I was just an invited speaker. At this seminar I was really looking forward to just sitting out there and listening, but at the last minute I was asked to give a presentation on "How Much Security Do You Need and How Much Will It Cost?" Several of the comments from Tuesday morning were quite interesting as I will discuss later, but among the more interesting was the question: "How much are you paying for security?" The panelists were not able to say. In that same spirit I will admit right up front that I am not going to be able to tell you what it is going to cost you for security. But I hope to give you some insight into ways to achieve adequate security at reasonable cost.

Before jumping into that though, I would like to comment on a book that I ran across by Norm Augustine, the Chairman of the Defense Science Board. It's called *Augustine's Law* and is a collection of his observations of how things have evolved in the military industrial complex. Apparently, he originally wrote the book as a serious paper, but nobody paid any attention to it so he rewrote it as a series of interesting vignettes that have very telling stories behind them. Read any paragraph and you are both amused and learn something.

He tells in the beginning of the book how back in 1909 the Army decided that it ought to buy one of these new fangled "aeroplanes" and it went through the normal procurement process. It put out an RFP, and got bids, and ordered a contract, and all that. It's interesting to note that the RFP was one page long. The bidder's had seven days to respond. The longest bid was four pages long. It took seven days to evaluate the bid. Thirty days after the contract was awarded, the Army had an airplane that worked. Clearly, we know something now that they didn't know about handling such matters. He then talks about the C5A, which is the world's largest aircraft and points out that a single copy of the documentation that was required to go along with the C5A wouldn't fit in the C5A. I highly recommend this book because it's full of interesting, amusing stories.

I had, originally, come here on Tuesday morning with a specific set of ideas I wanted to discuss. Some of them were poking at the idea that even though we now have this Evaluation Criteria which contains a range of acceptable solutions, there's a tendancy to always strive for the best you can possibly get. I wanted to argue for moderation, for accepting reasonable solutions before demanding perfect ones. I will make those arguments in a few minutes, but as I listened to some of the comments made here Tuesday morning I became concerned about a different set of ideas. So I am going to condense some of what I was going to say to squeeze in some comments about Tuesday morning.

First we heard that there are only a few organizations in the private sector that have a policy model like that of the Defense Department; the hierarchy of UNCLASSIFIED through TOP SECRET and compartments. I thought it was interesting that one of the organizations that does have a policy exactly like the DoD's is one that has no small

influence in the information handling business, namely, the IBM Corporation. I wanted to ask Bill Murray about that, but I guess he's gone. Contrary to Bill's comment, there are, I believe, a large number of organizations with a security policy model that is some adaptation of the lattice structure that the DoD is using.

We heard comments that the present physical and administrative controls are adequate, but they are not used effectively. I'm not going to stand here and argue that trusted computer systems are the answer. Computer security is a management problem and if management isn't serious about doing the physical, administrative, and procedural things that are required, trusted systems will be of very limited value. But when management is willing to do those other things, and wants to go to the next step, namely, protecting information from authorized users who shouldn't necessarily have access to everything on the system, then management must rely on trusted systems techniques.

We heard the comment that the use of the Criteria and a policy model like that of the Defense Department implies that you have a good information classification system, but that industry is in too much disarray to take advantage of these ideas. Well, I have a lot of trouble with that. Every organization has sensitive information of one kind or another. If it doesn't, it's not clear why it exists. Any organization that cannot identify its sensitive information is in real trouble. If your organization is in this condition, then trusted computer systems will not be of much help to you, and I worry about the future of your company. Certainly portions of your advanced planning or R&D results, or future strategies for acquisitions of other companies' strategies or whatever, has to be sensitive and it deserves some kind of special protection.

We are told that the real problem is that clerks are stealing money from accounts payable. Once again, I'm not going to argue that trusted computer systems are necessarily going to solve that problem. That was a problem that was there before there were computers. It's a problem that requires good management, accounting, and supervision for prevention, whether in an automated or manual system.

I will argue instead that there is a much bigger security problem rapidly emerging which few have thought about: namely, word processors. Everybody has his/her own wonderful little word processor. They can generate all this text, edit and change it, and print it out as a really good quality product very, very easily. There is no security problem because this system is hooked up right here in my own office. The next natural step is to link word processors together so that I can get the report that Sam's putting on his word processor without having to move a bunch of papers. So let's run a wire between them and while we're at it, let's link the whole building, and then extend the links to the other portions of the company or, maybe, other companies so we can share all that information.

Suddenly, the very limited vulnerability of that word processing system has changed dramatically. It used to be that a bad guy had to gain physical access to your machine. They don't have to do that anymore. Not only do they have powerful networks they can use to access your system, they have powerful computers that they can try to break in with. This problem concerns the protection of sensitive information with respect to the future planning of your company, a much more serious concern than clerks stealing from accounts payable.

We also heard the DoD characterized as highly organized and centralized as opposed to industry which is very fragmented and infighting and all that. I have to agree with Mario Tinto that I don't know what Defense Department that is. I worked for the Defense Department for 22 years and it must have been some other group they are talking about.

Now, dealing with the question of, "How much security do you need and what is it going to cost?" As a consequence of those comments, as a minimum, I believe that you must be able to identify which information is important to your organization and figure out how to label it according to its sensitivity. It's not clear how much it's going to cost to do this. The speakers on Tuesday morning indicated they weren't sure how to catagorize the levels of sensitive data. It is clear what it's going to cost if you do not label and protect your sensitive data. Without a clear way of knowing what your sensitive information is, your organization risks its future to a significant degree. If you don't automate information handling in your organization, you're going to have a tough time keeping up with your competitors. But if you don't have some way to isolate and protect your sensitive information on your automated system, then you run the risk of losing this information to anyone with any degree of access to your system. The only alternative is to keep all the important financial or future planning information on separate dedicated systems where it will not benefit from the technology advances that the rest of your routine and administrative services will undergo. You run the risk of having a very efficient and automated routine unsensitive administrative service and a manual and cumbersome capability for your highly sensitive requirements. That's not a good situation.

The 414's last summer showed that if you have a poor password system, you're leaving the door open and inviting thieves to come in, but passwords are not enough. Some of Tuesday's speakers were arguing that until we put good password measures into effect, we don't need all this other fancy trusted stuff. I won't disagree with that. If you leave the system maintenance account labeled SYSTEM with a password labeled SYSTEM (something that everybody knows since that's the way the manufacturer supplies it), then you deserve whatever is coming to you. My comments are directed to the people who are prepared to do those measures and want something else beyond.

I believe we are in for a really exciting time in the next few years. The Criteria - the lovely orange book - provides us with a very strong vehicle to move ahead *if* we do it in a reasonable and moderate manner, but I have a grave concern that, in spite of the fact that we have now identified a number of levels within a system, we will be repeatedly tempted to ask for too much. The risk is that we will push too hard for a level of system that we're not going to be able to get. We need to give the manufacturers a chance to evolve their products from what they currently have. No manufacturer is going to plunge in and introduce a new "System 36Ø" which is totally incompatible with what they've sold before. Many manufacturers have invested a

lot of money in very large development projects offering new and wonderful capabilities only to cancel them when they found they were incompatible with their current customer's base. We now have the Evaluation Criteria which can assist that evolution if we are careful. I do not want to claim that it's easy to evolve a major operating system, but we must give the manufacturers a chance. Several of them are moving down that path now.

The users also have to figure out what it means to have a trusted computer system. We've been operating for as long as we've had computers, in a dedicated mode in which everybody is cleared to have access to all the information on the machine. I know from my interactions with folks on many new programs that we don't really understand how to operate multi-level systems. We have to walk before we run.

For both those reasons, I want to argue that we should make as much use as we possibly can of the C-level system and of the lower B-level systems before we insist on A1's and beyond. I would like to give a couple of illustrations. The last time I was here, I talked about several programs that were going on in the Defense Department. One of these is the WWMCCS Information System, the modernization program for the Honeywell 6ØØØ's in our major command control systems. I was very pleased at what was evolving there because they were asking for a B level system, but were willing to initially start with something less as long as a path of evolution to B2 could be shown. This approach is not easy and runs the risk that the evolution may never happen, but I believe it is the right kind of strategy. The worry I have now is that the program will back further and further away from the B2 goal, postponing it indefinitely.

Another program that I discussed last time is the Interservice/Agency Ampe Program. This is the replacement program for the Defense Department's record message traffic system. It encompasses a very broad range of users; basically, the total range we have. Initially, the constraints on this system were too tough. It was to build a single system that could handle any level of classified traffic. Two years ago, the program was in real dissaray for a number of reasons. One of the major ones was that nobody could come up with a solution to the security problem so we came up with a compromise. We said, at least in the beginning, we will allow separate intelligence and general service classified systems. We will allow them to be on separate hardware. Eventually, we would like to go to a fully integrated system, but we don't have the technology to do that right now. That compromise allowed NSA to endorse the program that helped convince the Air Force to move ahead.

I wonder now, though, whether we don't still need yet another compromise. We are now asking for an A1 system, the best thing we know how to build. And we're asking for commercial off-the-shelf hardware. I fear that it's not going to work. Maybe we should look at another compromise that limits the unclassified aspects of the system just as the previous compromise relaxed the highest level security constraints. Maybe we ought to do something to limit the amount of unclassified use of the system. Again, the objective is to moderate the A1 requirement. Suppose we start with a B2 level system with an evolution path up to an A1.

I'm really concerned that we make effective use of these criteria and that we let the manufacturers understanding of how to build trusted systems and the users' understanding of how to use them grow in an orderly fashion.

I really believe that for us to get in a position where we have to build our own dedicated systems is really foolish. Take the AUTODIN system, for example. In the early 196Ø's that system was put together to handle the DoD's record message traffic. It is a good system. This year it is celebrating its twentieth anniversary. The trouble is that it is very much a dedicated system. It is not based on anything that is an on-going product so it is very hard to upgrade. They do not make germanium transistors anymore. We cannot get the parts for it. They do not make technicians who are excited by working on equipment of that vintage either, a much tougher problem. And so, we are faced with a problem of making some significant hardware emulations of those old machines so that we can get a few extra years of service. That is just one example. When the government has to build its own computer systems, that's the kind of problem we can get into. I think, instead of building our own special versions of these systems, we should be spending much more R&D money on building tools to allow us to trust the systems that others are building.

I have discovered that there is considerable interest in the commercial market in the Criteria and in trusted systems. The idea of making improvements to a system in an orderly, evolutionary manner is promoted by the Crtieria. The commercial view is that even modest enhancements can make a big difference. But the DoD's next steps in this area are crucial. If the DoD insists on systems that nobody knows how to build yet, then they are going to undercut this tendency. The DoD Computer Security Initiative can serve as a potent catalyst for promoting trusted systems, but only if it acts prudently.

This afternoon's session with the IEEE panel, looking at the Criteria as a standard in the general commercial world is an excellent idea and I strongly support it. If we are sensible about what we ask for initially and if we use these initial capabilites to learn how to make use of trusted systems, and if we give the manufacturers a chance to evolve their products, we are going to see a lot of progress in protecting sensitive information both in the government and in the private sector. But if we demand too much too soon and in doing so discourage the manufacturers and limit our own ability to understand how to use them, then the result is going to be, much as it has been, very few and very expensive systems that fall very short of the state-of-the-art.

Last week, I had the opportunity to attend the Computer Security Institute in New York City. There were a thousand people in attendence. General Faurer, the Director of NSA, gave the keynote speech which included an interesting presentation on the program of the Center and how the Center is trying to work with industry. He called for the establishment of a commercial consortium, allowing people in the commercial market to work with the Center to help evolve trusted systems. I don't know if the

IEEE group is a beginning of this. I hope it is. Later the same day, I had the chance to speak to a panel for the graduate program of the institute and I reiterated the commercial consortium idea. Later, Louise Becker chastized me, claiming you should never advocate those things if you don't know how to make them happen. I have thought about that quite a bit since then. I wish Louise were here because even if we do not know exactly how to do this yet, we need to get started. I would really like to work on such a commercial initiative and to encourage any of you who have thoughts about how to make that happen to join with me. Thank you very much.

FACTORS IN EVALUATING COMPUTER SECURITY

Zella Ruthberg

National Bureau of Standards

This panel will address security evaluations issues and their relationship to an activity called 'certification.' There is currently a new FIPS Publication 1Ø2, entitled "Guideline for Computer Security Certification and Accreditation," which was approved for release on September 27, 1983. This document gives guidance on how to establish a certification program and process within an organization so that sensitive systems and applications can be accredited for operation, as required by OMB Circular A71, TM 1. The key to performing a certification is the technical evaluation of the security of the system or application against its security requirements. It is therefore of interest to touch upon the major questions concerning computer security evaluations.

These major questions are: (1)What is being protected? (2) Why is it being protected? (3) How is it being protected? and (4) How well is it being protected? Each agency or organization carries out functions specified in its charter and must have information to carry out these functions. The elements that need protecting can be found within these information needs. Examples of information that clearly needs protection are tax records, social security benefits records, and criminal justice records.

The 'why' and 'how' of protection can be described in terms of The Control Network view of an agency/organization. The driving force for computer security needs come from agency/organization mission needs, Federal computer security policy, and user security needs. The computer security needs derived from these sources, together with top management's view of assets and risks provide the basis for agency/organization information control policy. Based on this information control policy, the agency/organization is then able to arrive at information control objectives (a form of information security requirements in the audit community) and statements about specific control technique objectives (called 'standards' in the audit community). Specific control techniques then implement this control policy.

The 'how well' of protection is at the center of computer security evaluation and concerns itself with measurement issues. These revolve around environment considerations, the need for evaluation criteria, and the forms of evaluation evidence (e.g., transaction flow data, logging and journaling, testing data, documentation, and interviews). Evaluation methods address this question.

FIPS PUB 1Ø2 discusses the four communities that perform such evaluations. They are: risk analysis, security safeguard evaluation (e.g., security officer activity), VV&T (verification, validation, and testing), and EDP audit. It should be noted that one can determine the security requirements from risk analysis, security safeguard evaluation methods, and EDP audits.

This, in brief, is the framework in which I view security evaluation and certification.

EVALUATION, THE DOD CERTIFICATION PROCESS, AND ITS RELATION TO THE TRUSTED COMPUTER SYSTEM SECURITY CRITERIA

William Neugent

The MITRE Corporation

My talk today is on factors in evaluating computer security. This presentation is from the DoD perspective. I'm not going to talk about how we do security evaluation in DoD. Instead I'm going to summarize some of the lessons that we've learned in doing security evaluation. The presentation will cover both the evaluation base line and the security evaluation process itself. First, let's examine the evaluation baseline.

EVALUATION BASELINE

The proper baseline for security evaluation is a good set of security requirements. There are three forms that requirements can take: policies, user requirements, functional and data requirements. I'm going to say a few words about each of these.

Policies

Policies are the first form that requirements can take. It has been my experience that in the DoD we sometimes forget about the laws on security requirements. For example, consider the Privacy Act.(1) Many people in the DoD assume that if you provide security, the privacy protection is inevitably included. That's just not the case. There are many of things that you have to do explicitly for privacy that are different from those things you do for security. For example, you have to allow people to examine and change stored personal data that pertains to them. You might have to maintain audit records for five years. You might have to record whether a person has given consent to have his or her personal information released. These functions are often not provided under the name of security. These are guidelines that identify the impact of the Privacy Act on computer systems. Two are Office of Management and Budget (OMB) Circular A-1Ø8 and Federal Information Processing Standards (FIPS) Publication 41.(2,3) Both are of help in assessing the impact of the Privacy Act. But the Privacy Act is only one of several laws that apply to DoD and affect the way we build systems. This has been and will continue to be an enormous help in formulating requirements. It requires adaptation for different application environments, but it's an excellent starting point. At the moment the Orange Book is an optional standard, unless you include it in a procurement. Other standards are mandatory. For example, where it is applicable, the Data Encryption Standard (DES) issued by the National Bureau of Standards (NBS) is the only unclassified encryption algorithm permitted for use by the Federal Government.(5) One thing to note about these standards is that even though they are external policies they tell you not only what to do but how to do it. Standards such as the DES thus impose implementation detail, which is necessary when a reliable method is found to perform a complex task.

Another form that policies can take is that of regulations. For example, DoD Directive 54ØØ.11 is the DoD implementation of the Privacy Act.(6) The computer security "Bible" for DoD is DoD Directive 52ØØ.28, which many of us have come to associate with Eugene Epperly.(7) Although many people are not aware of it, the military Services have their own implementations of the Directives. Much good work has been done on these Service policies. For example, consider the Service implementations of DoD Directive 52ØØ.28. First we have Air Force Regulation 3ØØ-8, soon to become Air Force Regulation 2Ø5.16.(8,9) This regulation includes a lengthy discussion on lifecycle security. Army Regulation 38Ø-38Ø is an excellent reference, with a useful security checklist and some insightful views on sensitivity levels.(1Ø) The Navy has produced a valuable policy in Office of the Chief of Naval Operations (OPNAV) Instruction 5239.1A.(11) The Marines also have their own computer security policy.(12)

These Service policies do not simply react to DoD-level policies but go beyond in cases where the DoD-level policy might not address a Service need. For example, most people think of the DoD security operating modes as including dedicated, system high, controlled, multi-level, and compartmented mode.(7,13) The Navy has added another called limited Automatic Data Processing (ADP) access security mode for those situations in which special access controls, as might be appropriate for proprietary information or other forms of For Official Use Only (FOUO) information.(11,14)

The Services are not the only organizations that have implemented DoD Directives such as 52ØØ.28. Some systems such as the Worldwide Military Command and Control System (WWMCCS) have their own implementation. The WWMCCS has Joint Chiefs of Staff (JCS) Publication 22 and several other documents that provide security policy and procedures.(15) The point here is that, whether or not these policy documents pertain to you, they contain much useful information and can serve as precedents and rules of thumb in formulating security requirements. So security requirements can be embodied in or derived from policies.

User Requirements

Security requirements can also be derived from user requirements. There are two types of user requirements: official and unofficial. Perhaps it seems optimistic to differentiate between official and unofficial user requirements when in so many cases we have no user requirements at all. Nevertheless, it's important to strive for user requirements that have been officially approved by responsible authorities. You cannot assume that because a user wants something his organization will buy it.

Recently I've been providing security support to the WWMCCS Information System (WIS) Joint Program Management Office (JPMO). The WIS has an extensive set of officially-approved user requirements. They are

packaged as Required Operational Capabilities (ROCs). Some of these ROCs are very general and not explicit about security. Instead, they describe the data that must be processed and the data flow involved.

Other WIS ROCs are very specific. For example, there's an Automated Message Handling (AMH) ROC that defines specific requirements for discretionary and mandatory access control, object reuse, labeling, trusted paths, reauthentication, and even for the system architecture.(16) some of the requirements are unusual. For example, when you review a message, you often annotate comments in the margin. The AMH ROC has a requirement to control access to these annotations, such that different annotations on the same message can have different access permissions associated with them. One ROC has a requirement for an automated form of two-man control in which two people must be present to access a file.

In addition to these official requirements, the WIS program also have unofficial requirements such as user surveys and an assortment of studies. These provide valuable context but they cannot replace the official requirements.

In the WIS program, the WIS JPMO has the responsibility to *interpret* user requirements. This involves resolving ambiguities, inconsistencies, and omissions and distinguishing ADP from non-ADP requirements. MITRE is assisting in this. In the security area we have taken many hundreds of user security requirements from the ROCs and reorganized them into functional groupings that were adapted from the feature and assurance requirements in the orange book.(4) While this helped to organize the user security requirements, it resulted in a 200-page document that contained, in addition to the many specific security requirements, a number of ambiguities, inconsistencies, repetitions and omissions. Therefore, we took the 200-pages of user security requirements and rewrote it into a much smaller and more integrated set of requirements, while preserving traceability to the original user requirements.(17) Furthermore, since full multi-level security, while a required goal for the WIS, is not yet commercially available, we found it necessary to identify subsets of the user requirements that apply to environments that are not fully multi-level secure. John Vasak and Chuck Youman of MITRE played the major role in this work. The point here is that official user requirements, as important as they are, do not necessarily provide the complete picture.

Functional and Data Requirements

The most detailed forms in which requirements are typically represented are those of functional and data requirements. Functional requirements can become more complex in a distributed environment. In a system such as the WIS, it's difficult enough just to determine which security functions need to be performed. Nevertheless, when the functions are distributed in a local area network (LAN) environment, it is also necessary to allocate the security functions among the distributed system components. This can be difficult for such functions as password management and audit data collection.(18)

In the past, the data requirements document has not been given sufficient attention within the DoD. This is the document that describes the data to be processed, its classifications and required authorizations, data sources and destinations, and requirements for data sharing and control. Data requirements become especially important as you evolve to different forms of multi-level security or to systems that include processors operating at different system high security levels.

Data requirements information can be difficult to obtain. For example, the WWMCCS has operated in a system high mode for many years. The WIS program is looking at ways to change that. Operating in a system high environment, however, WWMCCS users have not been required to examine the classification level associated with different WWMCCS functions. It's one thing to provide multi-level technology and quite another to start users thinking in terms of multi-level applications.

Time does not permit discussion of how to phrase requirements to facilitate their use as an evaluation baseline. Suffice it to say that this is one of several qualitative issues that mustbe kept in mind in defining requirements.

To sum up, security requirements are the baseline for security evaluation. Without them an evaluation is difficult, if not impossible.

SECURITY EVALUATION

The remainder of my talk addresses security evaluation. First, let us look at the objects examined in a security evaluation. There are two types of objects: development documentation and products.

There are several common problems in evaluating development documentation. One is insufficient security information. For example, development documentation often does not identify the security policy model to be employed, the security design principles to be followed, or the security analysis underlying critical design decisions. The orange book will be very helpful in solving this problem because it identifies the security information to be provided.(4)

Another common problem is out-of-date information. Even with effective configuration control, systems under developement are moving targets. This complicates the evaluation process and can make it inefficient and frustrating to the people involved. Inadequate identification of security functions is another common problem in evaluating development documentation. In the past, I've reviewed specifications for several trusted systems in which the specifications do not even identify the Trusted Computing Base (TCB). In other cases where the TCB is identified there is no definition of the degrees of trust involved. Sometimes security functions are identified but are neither adequately defined nor properly placed. In a system that has distributed security control, it's not sufficient to merely define the security functions allocated to each system component - there must also be a security architecture discussion that shows how the distributed security functions interact to achieve integrated protection for the system as a whole.

The final problem noted here is insufficient evaluation expertise in specialized areas. If you are responsible for evaluating a design document, you need to have someone assigned who has designed something similar in order to be

able to judge whether the design in question in reasonable. If the evaluation involves examining formal specifications or performing formal verification, highly specialized skills are involved. These skills must encompass not only formal methods in general but also the particular method involved, such as the Hierarchical Development Methodology (HDM) or the Formal Development Methodology (FDM).(19) These are extremely rare skills, so evaluation planning must budget time and money for training.

Next, let us look at some common problems in evaluating products. The primary evaluation activity of concern here is testing. One of the major problems is configuration management during testing. For example, when testing reveals a flaw, the flaw usually must be corrected. The configuration management process then must determine which areas need to be retested and whether new tests must be added to evaluate the correction. This can be difficult due to the "ripple effect" that such corrections can cause.

Another problem is determining when a product is not correctable. If a system reaches the latter phases of testing before this issue arises, then the evaluators have failed in their objective of detecting fatal flaws during the design phase. Nevertheless, for many reasons, it often becomes apparent during testing that a product has major uncorrectable security flaws. In these cases it is usually not possible to simply throw the product away. There is pressure from procurement offices, system program offices, and users who believe that a flawed product is better than none at all. Often the best solution to this dilemma is to remove especially vulnerable functions or reduce the threat by such measures as eliminating dial-up users or removing the most highly sensitive data.

A third problem is the provision of resources for independent internal testing. This can be illustrated with an example. Take the case of an Independent Validation and Verification (IV&V) contractor that must develop software to test the security interface between application programs and the operating system. This IV&V development effort requires a developmental system that is in many ways identical to the one being used by the development contractor whose software is being evaluated. This means that either a separate development facility must be employed or the IV&V contractor must use the development contractor's facility. Both approaches require substantial planning and have major pitfalls.

The final common problem noted in evaluating products is ensuring that penetration testing is properly used. Penetration testing is important and conveys a unique perspective on a system's strength. It can be difficult to make procurement offices understand, however, that the purpose of penetration testing is to form a confidence judgment on the system, not simply to "find and fix" flaws. The message that must be conveyed is that detection of certain numbers and types of flaws represents evidence that the system is not adequately protected. This cannot be corrected through correction of the particular flaws involved.

SUMMARY

Let me conclude today by summarizing some lessons I've learned in doing security evaluation over the years. First, and perhaps most important, is the need for accurate, complete, understandable security requirements. Without requirements, there is no reliable baseline against which to evaluate. This is an area in which the orange book will be of great assistance.

Secondly, coupling of the evaluation with development is essential. It is more efficient and far less expensive to find and correct problems when the required corrections are fairly simple and before people have a vested interest in an approach that's evolving.

Another lesson is the need for expert assistance in performing security evaluation. Without this assistance where it is required, the results from an evaluation are of little value. Furthermore, expert assistance is sometimes impossible to obtain, so provisions must be made for extensive training.

Aside from the highly specialized people, the overall quality of the security evaluation people is also important. The evaluation activity requires experience, discipline, and analytical skill.

Access to the people and documentation associated with the development effort is another requirement for security evaluation. In order to evaluate a system being developed, it is usually necessary to talk with the developers. They must take time from a busy schedule to make themselves available so that they can assist the people who will criticize their work. Similarly, evaluation often requires access to documentation that is not listed as officially deliverable under the contract. With both people and documentation, it's difficult to anticipate everything that will be required in an evaluation. Flexible interaction is called for. Such interaction is unlikely if there is an unfavorable relationship between developers and evaluators, and it is difficult to enforce flexible interaction through official contractual means.

Commitment of the approving authority is important. The approving authority is sometimes referred to as the accreditor or Designated Approving Authority (DAA). This authority is responsible for deciding whether a system provides acceptable security protection. It is necessary that the requirements and opinions of the authority be made known and issues resolved as early as possible in development.

Another lesson learned is that security needs an independent check. The normal development review process cannot be relied on to provide an adequate evaluation of security.

The final lesson is that evaluation cannot ensure a quality product. High quality evaluators and a thorough evaluation process are valuable, but evaluators do not build the system. The value of effective security evaluation is in finding and correcting problems early and in obtaining a reliable assessment of how much trust to place in a system.

Thank you.

REFERENCES

1. The Privacy Act of 1974.

2. Office of Management and Budget (OMB) Circular Number A-108, "Responsibilities for the Maintenance of Records About Individuals by Federal Agencies," 1 July 1975.

3. National Bureau of Standards, Federal Information Processing Standards (FIPS) Publication 41, *Computer Security Guidelines for Implementing the Privacy Act of 1974*, 30 May 1975.

4. DoD Computer Security Center, *DoD Trusted Computer System Evaluation Criteria*, CSC-STD-001-83, 15 August 1983.

5. National Bureau of Standards, FIPS PUB 46, *Data Encryption Standard*, January 1982.

6. DoD Directive 5400.11, "Department of Defense Privacy Program," 9 June 1982.

7. DoD Directive 5200.28, "Security Requirements for Automatic Data Processing Systems," 29 April 1978.

8. Air Force Regulation 300-8, *Automated Data Processing System (ADPS) Security Policy, Procedures, and Responsibilities*, 17 August 1979, to be superceded by Air Force Regulation 205-16.

9. Air Force Regulation 205-16, *Automatic Data Processing (ADP) Security Policy, Procedures, and Responsibilities*, soon to supercede Air Force Regulation 300-8.

10. Army Regulation 380-380, *Automated Systems Security*, 15 June 1979.

11. Office of the Chief of Naval Operations (OPNAV) Instruction 5239.1A, "Department of the Navy Automatic Data Processing Security Program," 3 August 1982.

12. Marine Corps Order P5510.14, *Marine Corps Automatic Data Processing (ADP) Security Manual*, 4 November 1982.

13. Defense Intelligence Agency Manual (DIAM) 50-4, "Security of Compartmented Computer Operations (U)," 24 June 1980, CONFIDENTIAL.

14. DoD Directive 5400.7, "DoD Freedom of Information Act Program," 24 April 1980.

15. Joint Chiefs of Staff Publication 22, *WWMCCS ADP System Security (WASS) Manual*, 2 March 1982.

16. *Automated Message Handling (AMH) Multi-Command Required Operational ls Capability (MROC 9-81)*, 26 May 1981, revised 5 July 1983.

17. J. M. Vasak, C. E. Youman, H. W. Neugent, *WIS ADP Security Requirements*, MITRE WP-84W00019, 6 January 1984.

18. WIS Joint Program Management Office, *Wis ADP Security Strategy*, DRAFT, September 1983.

19. M. H. Cheheyl, M. Gasser, G. A. Huff, J. K. Millen, *Secure System Specification and Verification: Survey of Methodologies*, MITRE MTR-3904, 20 February 1980.

SECURITY REQUIREMENTS, CONTROL OBJECTIVES, AND THE EVALUATION ROLE OF THE AUDITOR

Courtland Reeves

Peat, Marwick, Mitchell, & Co.

Zella Ruthberg has asked me to discuss the approach and concerns the independent auditor has on the issue of Security and Control within the Data Processing environment.

I will present to you the methodology used by Peat Marwick to perform Data Processing Security Evaluations (DPSE) and then comment on efforts being undertaken within major corporations to establish and implement security and security policies.

I have much material to present to you and wish to have you leave this presentation with sufficient information on how you might develop your own security risk assessment program. So, I ask you to bear with me as I take you on a jet-ride through the numerous slides I will be presenting.

During my presentation you will note that the approach the non-government community is taking towards implementing security and controls is very similar to what you will hear during this conference. The major differences, if any, will be the degree to which the non-government community has been successful in codifying and implementing security within their respective organizations.

However, given time and sufficient effort, we believe that both the government and non-government organizations will obtain the same product - a secured environment with an annual security objective and plan to measure security progress.

Back in the early 197Ø's, Peat Marwick faced up to the issue of many company's growing dependency upon data processing. We noticed that:

(Slide 1) New computer capabilities were being announced almost every day and that it was inevitable that on the one hand, the users were going to want to take advantage of the new technology and on the other hand, the people responsible for making it work would be hard pressed to keep up with the demand.

(Slide 2) As the technology became more complex, the data processing people began to develop their own language and, in many cases began to live somewhat isolated from other elements within the organization.

(Slide 3,4) Organizations began to rely heavily on the computer for carrying out their day-to-day business tasks. More and more information was being stored in the computer to the extent that, in many cases, most of an organization's data was becoming computerized and could only be effectively accessed and utilized if the computer environment continued to function effectively.

(Slide 5) And, the potential for misuse or abuse of data and information was becoming much greater with the use of the computer.

As a result of these issues, we recognized that if we were to carry out our audit function for our clients, we had to equip ourselves with the skills and an approach that would allow us to evaluate security in a complex data-processing environment. We developed a set of guidelines for evaluating data-processing security.

These guidelines are embodied in a large manual which consists essentially of several thousand questions which can and should be asked in order to determine the degree which the environment is secure.

(Slide 6) Security means different things to different people. In preparing these guidelines, our perspective on the evaluation of security was to determine the degree of risk associated with the way in which the data-processing activities are planned, managed and carried out. In other words, it is an attempt to determine, for management, how vulnerable the organization is to loss of its data-processing capability. This involves determing the impact such a loss would have on the organization as well as the probability of the loss occurring.

(Slides 7,8) Because of the complexity and its degree of interaction and overlap with other areas of an organization, it is necessary to segment the data-processing environment to provide a structured means of looking at it in a meaningful way. In the security analysis program (DPSE), we arbitrarily segmented the environment into ten elements.

Applications appear at the core of the environment and we think this is appropriate since applications are the basic reason for the existence of the data processing environment. In the final analysis, the applications are what must be kept secure, including the data they contain and the instruction programs whereby they are carried out.

Surrounding the applications are the segments of the data-processing environment itself: Processing, Communications Systems, Data Base, and operating and other software. Naturally, not all of this will exist in every environment.

The next segment is the physical environment in which the data processing functions are carried out. Surrounding this are the standards which have been established to govern the maintenance of security in the data processing environment, as well as the internal audit function which should ensure that the standards are followed.

Finally, numerous administrative activities are involved to ensure that the organization not only functions well but continues to function; in other words, that it is not unreasonably vulnerable to disaster, mismanagement, sabotage, or other conditions which may result in loss of accurate and/or continued operations.

(Slide 9) The Security Analysis (DPSE) program has three main objectives:

The First objective is to evaluate the extent and adequacy of security in the data-processing area and to develop practical recommendations for improvement where it is indicated. The operative word here is practical; control for control's sake is not only wasteful but can be counter-productive in its impact on the data-processing organization. One must be careful not to spend a dollar to control a dime but to ensure that standards, controls, and other aspects of security recommendations are indeed necessary and that they are seen to be necessary by those responsible for efficiency of operation. Controls are an overhead which, to some degree detract from efficiency, which has been a long standing contention between auditors and data processors.

The Second objective of the program is to evaluate if and how the Security Analysis/DPSE approach could be applicable in other areas of the organization beyond data-processing. The Security Analysis/DPSE is basically a highly structured approach to evaluating risk and it is largely the terminology which orients it specifically to data-processing. If the terminology is changed the approach may well have applicability in other areas.

The Third program objective is to put in place within the organization the capability to carry on the Security Analysis/DPSE program in the future. A program such as this cannot be a snapshot. As such, it would be history immediately after the picture was taken. Rather it is necessary to have an on-going review, not only to ensure that controls continue to exist and be adhered to, but also to ensure that the controls continue to be necessary and appropriate.

SCOPE OF THE SECURITY ANALYSIS/DPSE PROGRAM

(Slide 10) The Security Analysis/DPSE program or guidelines checklist is organized into ten segments. Within each of the ten segments, there are a number of sections within each of which are numerous questions that can be addressed to evaluate security in any one of the sections. The Security Analysis/DPSE program has three main phases:

-Phase I is essentially the development of an understanding of the particular data processing environment, in effect to develop a profile of the organization, the security problems which potentially exist, and the risk associated with each of these areas. Out of that profiling, it is possible to determine which segments of the DPSE program need to be addressed in order to investigate the highest priority areas.

-Phase II involves tailoring the Guidelines to the specific environment, carrying out the evaluation of security, analyzing results, and meeting with management to consider alternatives for appropriate corrective action.

-Phase III consists of developing final recommendations and preparing a written report.

(Slide 11) In *Phase I*, the first problem is to define the problem and the risk. This is done by gaining familiarity with the data processing environment, carrying out an initial analysis of the internal audit and EDP Division, as well as users, and identification of major areas of risk priority for attention. Out of this, a problem or risk profile is developed.

This EDP profile usually contains a list of the problem or risk areas on the left hand side of the matrix and the functional areas of the organization across the top. In this way, we are able to indicate which functional area is affected by a problem and to what degree, i.e. high, medium, low, or not at all.

An example of the information that might be contained within the EDP Profile matrix could include the following:

PROBLEM/RISK	SCOPE OF CONCERN				
	Data Center	Remote Location	Systems Development	Internal Audit	ETC.
On Line Controls Unsatisfactory	H	L	H	M	—
No EDP Group Responsible for Controls	M	—	M	H	—
Potential loss Due to Unauthorized Access of data	M	H	—	—	—
Inadequate Back-up And Recovery Procedures	H	H	—	—	—

The next step is to develop a similiar matrix for each Segment by plotting the major problems or risks against the DPSE segments; on this matrix, the same problems or risk elements are listed down the left side, but this time are plotted against the ten Segments. An example of DPSE Segment Profile by the functional components might include the following:

PROBLEM/RISK	SCOPE OF CONCERN					
	Internal Audit	Admin	Physical	Stan-dards	Pro-cessing	ETC.
On line controls Unsatisfactory	—	—	—	H	H	—
No EDP Group Responsible for Controls	H	—	—	M	H	—
Potential Loss Due to Unauthorized Access of data	M	—	—	L	H	—
Inadequate Back-up Recovery Procedures	—	M	—	H	H	—

With these two matrices, we have developed a clear picture of:

- those problems or risk elements that need to be considered
- the organizational component that needs to be reviewed to evaluate the individual problem
- the segment of the DPSE program which contains the questions appropriate to each problem.

Given this information, it is now possible to develop a work plan for Phase II. This is then discussed with management to ensure that it reflects their concern and priorities and also to make them aware of the detailed Phase II program.

(Slide 12) *Phase II* is the detailed security evaluation study, and begins with the tailoring or development of the guidelines to reflect the problems and risk priorities that have been established in Phase I.

We tailor the program by taking the Data Processing Security Evaluation (DPSE) Guide and selecting from it relevant Segments appropirate to the areas to be reviewed, modifying terminology to suit the particular data-processing or corporate environment and, in fact, creating a new program or checklist.

Typically, we have found that the tailored guidelines usually consist of some twelve to fourteen hundred questions spread among the ten Segments. With this program, we address the existing controls and evaluate the adequacy of the security and control procedures, which results in plans for carrying out compliance and deficiency tests. The high number of questions is usually due to the fact that each one of the Segments will have various Sections.

After the Segments (i.e., Administration, Operating System Security, Data Base Security, Communications Security, etc.) are selected and the Sections for each Segment are defined, a detailed workplan is developed for each Section and its Subsection.

An example of a detailed work plan for the Communications Security *Segment* could include the following *Sections*:

- Network Analysis
- Code Controls
- Cryptography and Data Transmission
- Off-Line Communications
- IMS Communications
- Terminals
- CICS/VS
- Timesharing
- Access Methods
- CMS

Each one of these Sections would have various Subsections, where each of the Subsections would be detailed by Scope, Objectives, Risk Evaluation, Positions to Contact, and Substantiation tests to be used in the compliance and deficiency testing. As the description implies, compliance tests are carried out to ensure that where controls are in existence, they are being complied with or enforced. Deficiency tests are designed to determine whether the control is adequate for its purpose.

An example of a breakdown for the Communications Security Segment for the Code Controls Section could include Subsections such as Passwords, Lockwords, Security Tables, and Personal Identification.

Once the Sections to be included within the study have been selected, the reviewing team applies a weighted value to each Section and Segment. This weighted value emphasizes those Segments and Sections which are more important than the other areas being reviewed.

An example of weighting each of the *Sections* within the Communications Security *Segment* could be as follows:

Communications Security Segment
Suggested Segment weight = 15/1ØØ

Suggested Section Weights are as Follows:

-Network Analysis	4
-Terminals	3
-Code Controls	6
-Cryptography and Data Communications	3
-CICS	4
-Timesharing	Ø
-ETC	
Total Segment Score	2Ø

The quantification of the results of the review will provide the review team with a total Section and Segment score upon which to measure current and future progress towards improving both Segment and organizational security.

The work program for each of these Subsections would include developing questions to analyze the level and adequacy of security within the organization.

As an example, the Subsection "Passwords" has the objective to determine if the organization has adequate password control procedures in practice to protect usage of the system. Questions, such as the following, are structured to answer this objective:

> Is a user-oriented password system utilized?
>
> Is the control over issuance and use of password effective?
>
> Are all passwords changed and reissued at least semi-annually, and are passwords immediately deactivated upon termination or deauthorization of their holder?
>
> Do procedures exist whereby a user can have his password immediately invalidated and a new one issued when he believes secrecy may have been compromised?
>
> If passwords are issued on cards which the user carries, are the cards void of any information indicating that they pertain to a computer system?

Each question is assigned a value, depending upon its importance to the control/security needs, and points, up to the assigned value, are awarded depending upon the findings.

Once the reviewer has completed the Subsection review, all items/questions values are summed to produce a total score for the reviewed Subsection. The mean value for the Subsection is transferred to a Quantitative Summary sheet where each Subsections score will be tabulated, weighted, and summed to produce a Segment total. The Segment totals are then multiplied by the Segment weight to determine each Segment's score.

All Segment scores are summed and divided by the total Segment Weight (1ØØ). The resulting score, representing the mean value of all Segments, provides the organization with an indicator on how the organization is performing to meet its security requirements and objectives.

By carrying out the security evaluation program and by completing the compliance and deficiency tests, we have, in effect, developed a very comprehensive, formal, highly structured, detailed set of working papers. Not only does this provide support for development of recommendations

for improvement, but it provides the framework for establishing the program as an ongoing activity within the organization.

The results of the program are analyzed, recommendations developed and presented to management for discussion and agreement.

(Slide 13) Phase III of the program consists primarily of finalizing the recommendations following discussions with management and development of the final report.

The final report usually takes the form of two volumes: Volume I providing a management perspective on the program, its findings, major areas of recommendations, resource needs, policy, standards and timing and cost implications for implementing improved security and control methodologies; and Volume II containing a detailed analysis of the specific findings, cross-references to detailed working papers and specific recommended improvement actions.

The outputs which can be expected from a program such as this are:

-An analysis of the level of security that exists in key high-risk portions of the data-processing environment.

-A definition of security and control problem risk areas as discussed and agreed with management, and practical recommendations for improvement.

-A comprehensive security and control program that is an ongoing program as mentioned earlier, and takes account not only of the procedures, but also the organizational policy considerations necessary to make it effective.

-The development and acceptance of an ongoing security and control work program as an integral part of the internal audit and data-processing procedure.

BENEFITS OF THE SECURITY ANALYSIS/DPSE PROGRAM:

(Slide 14) A set of detailed recommendations in order of priority, for improvement in those areas which have been reviewed.

A structured methodology for an ongoing evaluation program.

The development of a control strategy which will define the roles of responsibilities for both the EDP and Internal Audit organization, organizational implications, resource requirements necessary to carry on continued evaluation and, most importantly, ensure adequate security and control is maintained.

(Slide 15) This curve portrays what one expects to find as the relationship between the existing level of control and the importance of control in any given application or function. Where the control is less important due to lower risk/vulnerability, a lower level of control is required. As the importance grows, so must the level of control. This is not a straight line growth, but rather, it increases exponentially as shown by the shape of the curve.

What we usually find in organizations where we have carried out the review is that the curve is somewhat flatter than expected. In the areas of less importance, the level of control is usually adequate; this is because the control is usually simpler and easier for all parties to accept and maintain. As the importance of control grows, however, which it usually will where the applications become more critical such as in an on-line system, distributed processing, complex data bases, and critical business applications, the control itself becomes more complex and difficult to plan and exercise.

The area between the two curves represents critical area or degree of risk, and our objective should be to reduce that risk by introducing more effective security and control procedures, thereby bringing the bottom line more in relation to the top.

In the future, we will inevitably introduce more complex and more critical applications which have an increasingly higher importance of control. Unless appropriate security and control procedures and policies are in place, we run the risk of having the critical area of risk grow larger, seriously increasing the vulnerability of the entire organization.

CLOSING REMARKS

Recently, I have performed security reviews of several large non-government organizations. These organizations have included a major airline, major financial institutions, an insurance agency and a brokerage house. What I am finding is that each one of these organizations is in a different stage of developing and implementing security.

Organizations can implement security either horizontally or vertically.

- horizontal implementation is the process whereby an organization establishes a corporate-wide security policy, develops procedures, and requires all divisions to implement security procedures to control and protect corporate data/resources.

- verticle implementation is the process whereby an organization will implement security within a specific division, i.e., Data Processing, in an effort to control and protect corporate/data/resources.

The obvious benefit of Horizontal implementation is evident in that all components of the organization must work together to meet the corporate requirement to protect corporate data, whether it be in the form of computer printout, fiche, terminal screen, bond paper, etc.

The major deficience in Verticle implementation, is that the responsibility for implementing security rests with a single department, but places little responsibility within the other departments for securing their own data or the data produced by the protected environment.

As a result, I find that the Data Processing departments are very security conscious, but that the Users of the Data Center, not only resent control but are negatively disposed towards the needs of the data processing departments requirements to implement reasonable data security.

The organizations that I have reviewed, for the most part, began and are continuing to implement security Vertically.

What is interesting, is some of these organizations are finding out that Verticle implementation does not provide the environment to adequately meet the corporation's security requirements.

As a result, some of the major organizations are now establishing corporate-wide security committees whose responsibility is to develop Corporate-Wide Security Policies and Procedures.

This new effort by these major corporations is an attempt to recognize the fact that security is a corporate-wide issue and is no longer solvable only by data processing. Security must be implemented across all major corporate components, not just within data processing in order to be successful.

I want to thank Zella Ruthberg for asking me to speak to you today, and thank you for giving me your time during this conference.

I have enjoyed meeting with you today, and hope that you will be able to use some of the information I have presented in developing your own security analysis programs.

Thank you.

The technology is changing so frequently that new generations of equipment are appearing every few years -- rapid obsolescence is the result.

Slide 1

A new group of specialists has developed around the technology -- with its own language, jargon, attitudes, and management style.

Slide 2

Non data-processing areas of business have become more and more alienated from the data-processing area -- mainly due to lack of understanding of what **EDP** really is and how it works.

Slide 3

The data stored within these large, computerized systems now constitute the majority of the organization's information base.

Slide 4

The incidence of "white-collar" crime such as fraud or embezzlement has increased dramatically in recent years, much of it aided by computer technology.

Slide 5

SECURITY EVALUATION

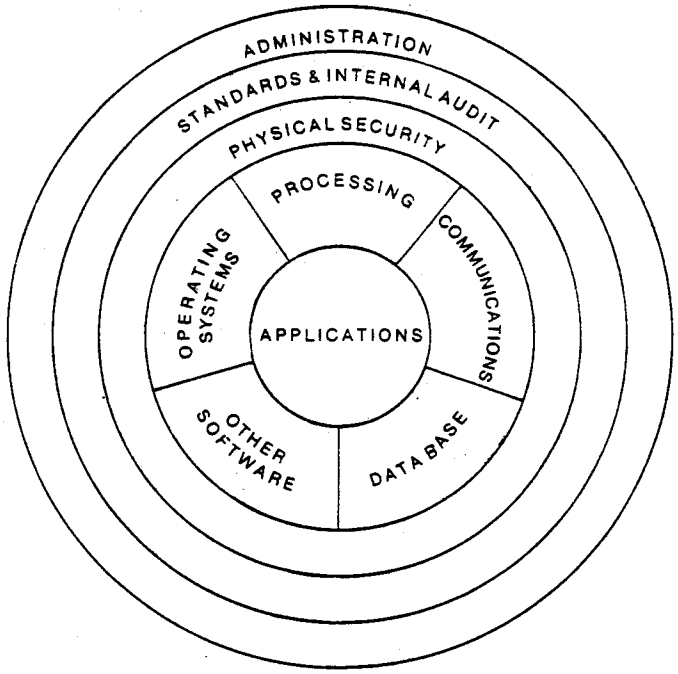
Evaluation of the degree of risk associated with the way in which data-processing activities are planned, managed, and carried out.

HOW VULNERABLE ARE WE?

Slide 6

THE EDP ENVIRONMENT

A SECURITY PERSPECTIVE



Slide 7

Scope of the data processing security evaluation program

SUMMARY

Internal Audit	Operating System
Administrative	Communications
Physical	Software
Standards	Applications
Processing	Data Base

Slide 8

DATA PROCESSING SECURITY EVALUATION

PROGRAM OBJECTIVES

- evaluate extent and adequacy of security in the EDP area and formulate practical recommendations for improvement.

- evaluate applicability of DPSE approach for evaluating risk in other areas of the organization.

- develop the organizational requirements and the assignment of responsibility for continuing the evaluation of security in the EDP area and in other areas of the organization.

Slide 9

SCOPE OF THE DATA PROCESSING SECURITY EVALUATION PROGRAM

DETAILS

INTERNAL AUDIT
- STAFFING AND ORGANIZATION
- BUDGET AND ASSIGNMENTS
- CHARTER AND OBJECTIVES
- WORK PROGRAM AND SCOPE
- DOCUMENTATION
- DISASTER RECOVERY PLAN

ADMINISTRATIVE
- EDP ORGANIZATION & PERSONNEL
- EDP PLANNING
- DEVELOPMENT AND ACQUISITION
- RESOURCE MANAGEMENT
- EDP LEGAL CONCERNS

PHYSICAL
- DATA CENTER SECURITY
- DATA PROCESSING SUPPLIES
- DATA SECURITY
- EQUIPMENT SECURITY
- DOCUMENTATION PROTECTION

STANDARDS
- APPLICATION DESIGN
- PROGRAMMING
- DATA CENTER OPERATIONS
- OPERATING SYSTEMS
- COMMUNICATIONS
- DATA BASE MANAGEMENT
- USER INTERFACE

PROCESSING
- OPERATIONS ENVIRONMENT
- DATA SECURITY
- PRODUCTION
- EQUIPMENT
- OFF-SITE PROCESSING

OPERATING SYSTEM
- GENERAL CONSIDERATIONS
- MECHANICS/FUNCTIONS
- MANAGEMENT CONTROL
- EFFECTIVITY AND INTEGRATION
- SPECIFIC OPERATING SYSTEM

COMMUNICATIONS
- NETWORK ANALYSIS
- TERMINALS
- CODE CONTROLS
- CRYPTOGRAPHY AND DATA TRANSMISSION
- SPECIFIC NETWORK HANDLER

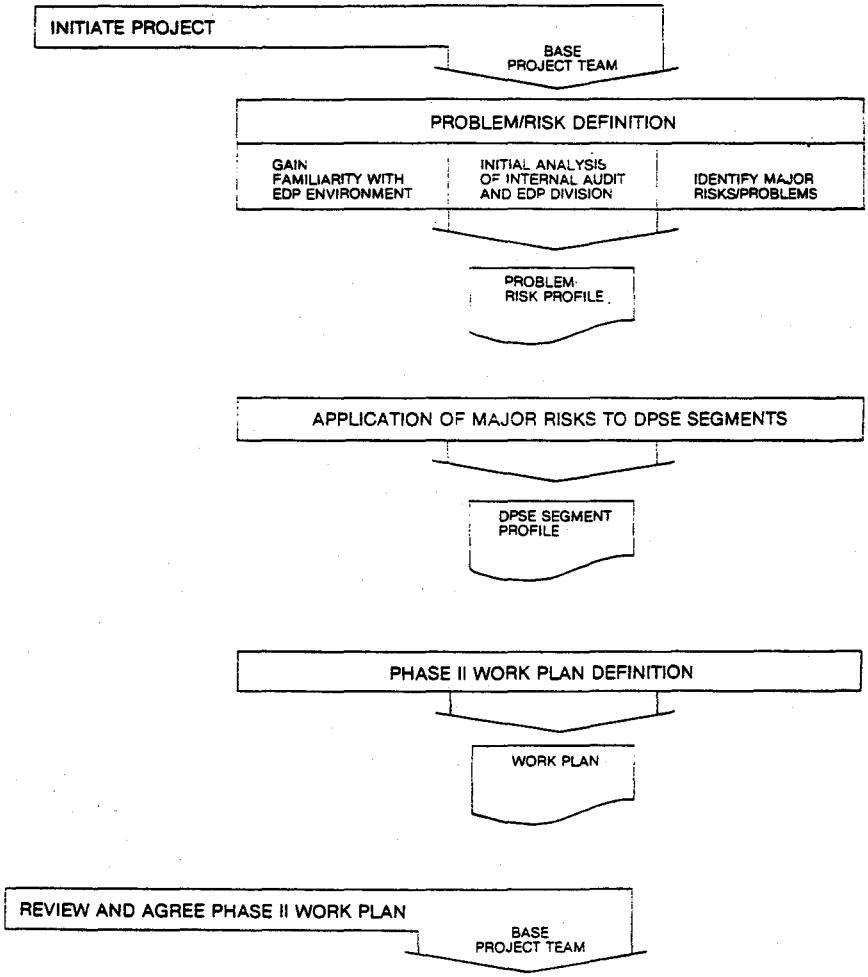
DATA BASE
- GENERAL CONSIDERATIONS
- SPECIFIC DBMS

SOFTWARE
- ASSEMBLERS AND COMPILERS
- PROGRAMMING LANGUAGE CONTROLS
- PRIVATE & TEMPORARY LIBRARIES
- SOURCE & PROCEDURE LIBRARIES

APPLICATIONS
- DOCUMENTATION
- DESIGN TECHNIQUES
- SYSTEMS ACCEPTANCE
- PROCESSING

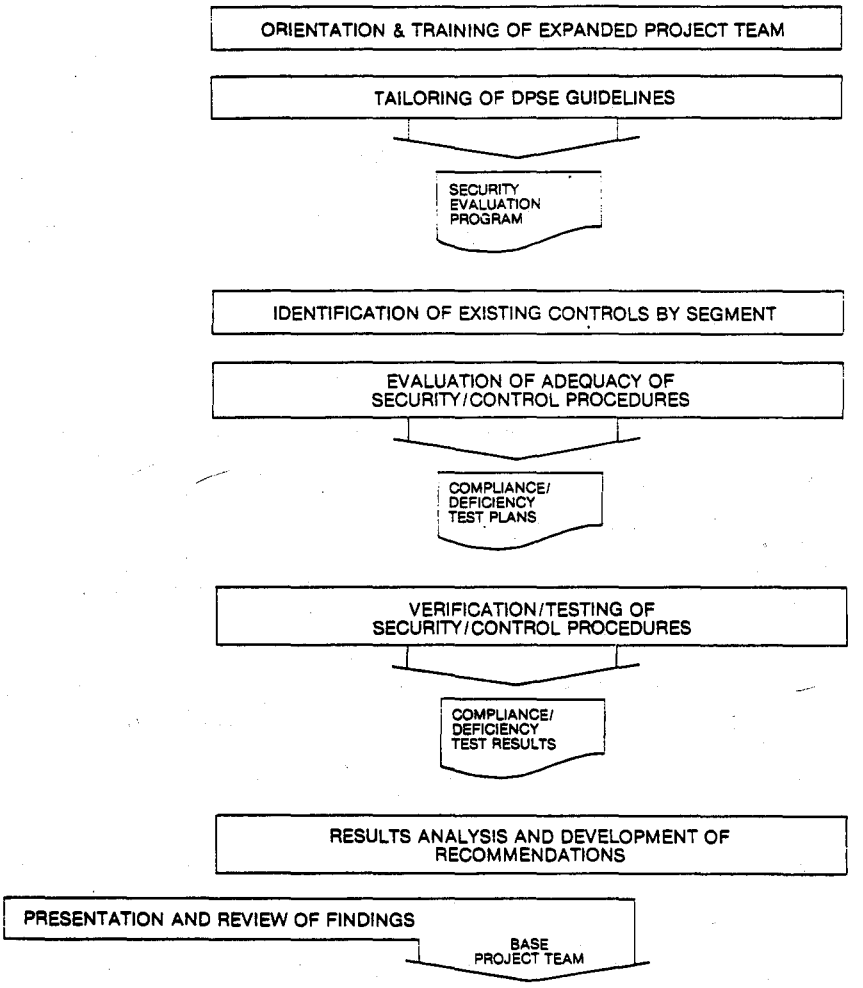
Slide 10

DPSE PHASE I WORK PLAN



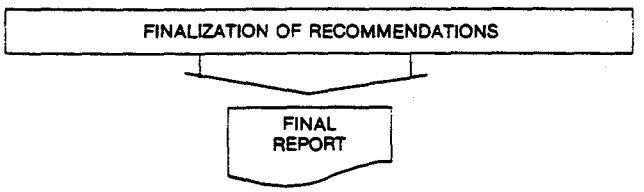
Slide 11

DPSE PHASE II WORK PLAN



Slide 12

DPSE PHASE III WORK PLAN



Slide 13

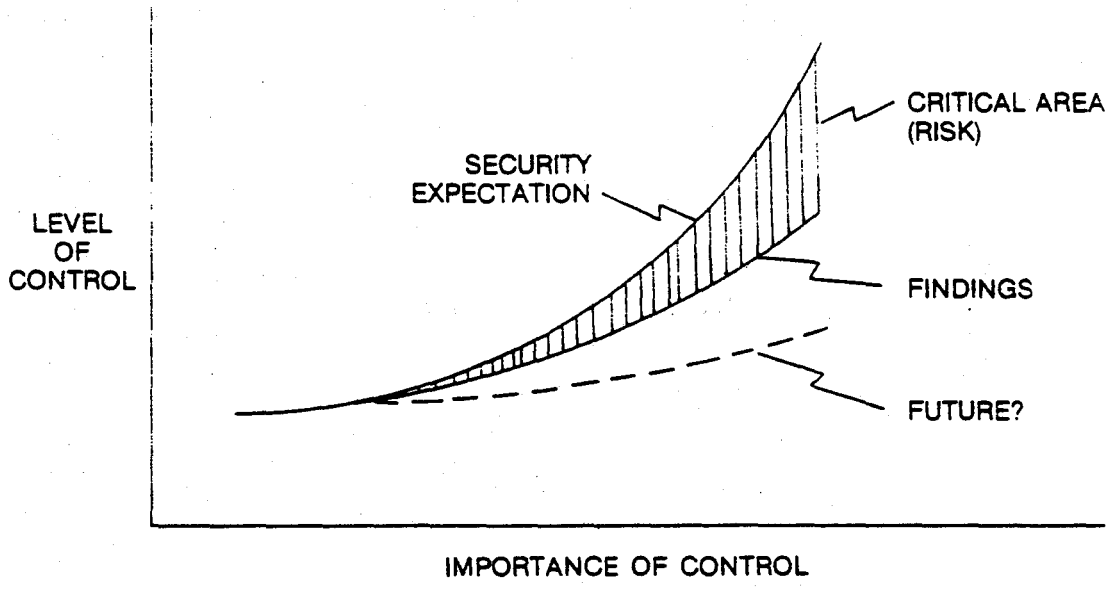
BENEFITS OF PROGRAM

- set of detailed recommendations for improvement, in order of priority
- structured methodology for an ongoing evaluation program
- development of in-house expertise
- development of a control strategy
  - -- EDP and internal audit roles and responsibilities
  - -- organizational implications
  - -- resources

Slide 14

EDP SECURITY

AN IMPRESSION



Slide 15

CIVILIAN AGENCY VIEWS OF EVALUATION, CERTIFICATION AND ACCREDITATION

Lillian Duffey

Federal Emergency Management Agency

I was asked to speak on factors in evaluating security and so I'm going to concentrate on those factors. The first factor I have considered is the agency mission. What do we have to do? As you read that you will see that there are two times when we have to perform our mission - our mission is divided into two time segments. Times of emergencies when we have a protection mission - to preserve the continuity of our constitutional government, and other times. That would be the primary factor in evaluating your system's security. A second factor would be the sensitivity levels of your system resources. As you can see from this slide our sensitivity levels have quite a large range - from compartmented national intelligence down to maybe a merit award in payroll. Another point that I would like to make is that the controls on our resources are also sensitive. We expect the controls to be protected as well as the resources themselves.

When you are asked to evaluate you want to know what you are evaluating. This is what we've considered to be an automated information system. We're evaluating more than just hardware and software. We have to consider people, procedures they follow or don't follow and the word processors and supporting telecommunications.

Another factor to consider is what do we mean by security? As you know there is no definition for automated information systems security that people agree on but to have a program, we've adopted this definition. It is protection against losses from unauthorized activities that could be deliberate or unintentional.

The responsibilities of the director for security comes from Transmittal Memorandum No. 1 which assigns responsibilities to executive agency heads. We take a broad definition of security and it's derived from this second bullet. If you look at your orange book on page 7Ø, you will see that the first bullet is mentioned but assuring that automated processes operate effectively and accurately is not mentioned. However, TM1 assigns that responsibility to the agency head so we have to consider that. There is another definition, probably, for this and that's quality assurance. So this second bullet motivates a lot of our security program.

To fulfill his responsibilities, the director of FEMA issued a FEMA-wide instruction. It does the two things that are required. It establishes basic security policy and assigns responsibility for a computer security program required by TM1 - which we have extended to automated information systems. The Office of Information Resources Management is assigned that responsibility. The policy itself is essentially the same as DoD policy. And I'm not referring to the policy that is at the star property level. I'm talking about what you will find in 52ØØ.28. We have adopted that policy because a lot of our functions are similar to those of DoD in times of emergencies and one of the things that you should do in evaluating security is to determine whether a system adheres to this policy.

At this point only the instruction has been approved, but we have a security program documented. We hope to implement it as soon as we have gone through all the review and comment process, and approval of the final draft. The program follows the minimum security program required by OMB. As you can see these are the seven points that are required by OMB. We are considering risk analysis as a necessary management tool and it will be done under the three other bullets of this program, installation management, application management control process and in acquisition/ operation specifications.

This is how we propose to implement this. We look at automated information systems as composed of the processing systems themselves and the application systems. We are assigning responsibilities for security not only to information resources management, but to the program offices as well, and that covers the word processors that are in the various offices. IRM will have responsibility for the networks. Right now most of our network is a set of disjoint nodes, but as we progress toward a single network these automated information systems that we have now will progress towards a single automated information system. If you look across the board we are covering the first requirement, installation security, with the evaluations done by these people, and going across through the management control process through the certification and acreditation program - in all of these the program offices are involved. The final dollar amount that is spent on application systems security is going to be approved by the program offices. That also shifts responsibility to the program offices having responsibility for the applications - where it should be. They have the authority to spend the money. This places responsibility for applications systems security with those who have authority to commit funds to pay for it and to manage the people who are using the systems.

The factors I have mentioned are the broad factors we consider in evaluating the security of FEMA automated information systems, and these factors are consistent with our philosophy of information resources management.

FEMA MISSION

Under the direction of the President, protect the civilian population and resources of the Nation and preserve the continuity of constitutional government in time of emergencies. Develop programs and activities for preparedness for, mitigation of, response to and recovery from natural, accidental, terrorist, and wartime civil emergencies.

Slide 1

SYSTEM RESOURCES SENSITIVITY LEVELS

LEVEL

6 Compartmented national intelligence, SIOP - ESI, life critical, or system controls to this level

5 Top Secret, extremely private or proprietary, or system controls to this level

4 Secret, or system controls to this level

3 Confidential, vital to FEMA mission, high privacy or proprietary, or system controls to this level

2 Large dollar volumes (1Ø million dollars/annum or higher) or system controls to this level

1 Lower dollar volumes, software development, or system controls to this level

Slide 2

WHAT IS AN AUTOMATED INFORMATION SYSTEM (AIS)?

AN INTERACTING ASSEMBLY OF

- Procedures

- Processes

- Methods

- Personnel

- Equipment
  - automatic data processing
  - word processing
  - supporting telecommunications

Slide 3

WHAT IS AIS SECURITY?

PROTECTION AGAINST LOSSES FROM UNAUTHORIZED

- Disclosure
- Modification
- Destruction
- Denial of service

Slide 4

RESPONSIBILITY OF THE DIRECTOR

RESPONSIBILITY OF THE DIRECTOR INCLUDES:

- Establishing safeguards
  - physical
  - administrative
  - technical
- Assuring that automated processes operate
  - effectively
  - accurately

Slide 5

FEMA INSTRUCTION 1500.1

FEMA INSTRUCTION 1500.1

- Establishes basic AIS security policy
- Assigns responsibility for the
  - development
  - implementation
  - and operation of

a FEMA AIS security program

Slide 6

FEMA AUTOMATED INFORMATION SYSTEMS SECURITY POLICY

Individual Accountability

Environment Control

System Stability

Data Integrity

System Reliability

Appropriately Secure
Communications Links

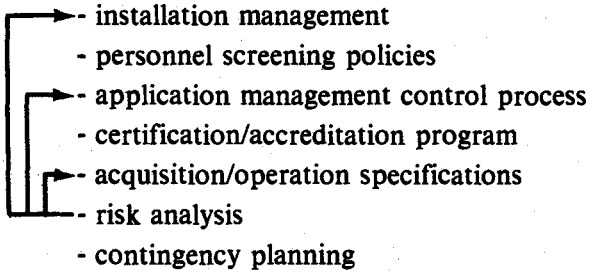
Appropriate handling and storage
of Sensitive Information

Slide 7

PROGRAM IMPLEMENTATION

When documented program is approved IRM will begin

implementing a distributed AIS security program



Slide 8

AUTOMATED INFORMATION SYSTEM

AUTOMATED INFORMATION PROCESSING SYSTEM (S)			COMPUTER APPLICATION SYSTEM (S)			
COMPUTER INSTALLATIONS		NETWORKS	MANAGEMENT CONTROL PROCESS		CERTIFICATION/ ACCREDITATION PROGRAM	
IRM	PROGRAM OFFICES	IRM	IRM	PROGRAM OFFICES	IRM	PROGRAM OFFICES
INSTALLATION MANAGER INSTALLATION SECURITY OFFICERS TERMINAL SECURITY OFFICER	PROGRAM MANAGERS INSTALLATION SECURITY OFFICERS TERMINAL SECURITY OFFICER	INSTALLATION MANAGER NETWORK SECURITY OFFICER	APPLICATION RESOURCES MANAGER	APPLICATION PROGRAM MANAGER	CERTIFICATION PROGRAM MANAGER APPLICATION CERTIFICATION MANAGER TECHNICAL EVALUATORS	APPLICATION PROGRAM MANAGER ACCREDITING OFFICIAL

Slide 9

CIVILIAN AGENCY VIEWS OF EVALUATION, CERTIFICATION AND ACCREDITATION

Fred Tompkins

NASA

First of all, I should like to indicate that I am speaking on behalf of Mr. Russell Rice, the Computer Security Program Manager for NASA Headquarters. I think it's important to contrast what I am going to present to you today with the remarks of the previous speakers on this panel. Court Reeves gave you the auditor's viewpoint and approach in evaluating the security of a data center, Lillian Duffey presented an overview of the computer security program being implemented at FEMA, and Bill Neugent looked at certification and accreditation from the DoD perspective. In NASA, the current evaluation and certification guidance has been directed toward existing applications software. I should also like to note at this time, that at NASA there has been a very deliberate program to develop guidelines for the NASA centers on very specific topics and areas of the overall computer security program. It is important to understand the NASA environment with respect to how NASA Headquarters views its relationship with the NASA Centers. NASA Headquarters issues policy and publishes guidance on methodoligies for implementing policy. Management and procedural implementation is left to the Centers. In many senses, NASA Headquarters operates as a corporate headquarters with centers operating as wholly owned subsidiaries.

The set of guidelines I want to discuss today is directed at existing systems only. While the basic approach is useful in looking at new systems evaluation and certification, this was not the focus of this document. Before we get the process of evaluation and certification for existing systems, it is important to know the assumptions and contraints that were established prior to developing the approach. First, in the case of existing systems, there are only two options relative to certification. Option one is to certify the safeguards as being appropriate to, and adequate for, the application or to certify with qualification pending the implementation of additional safeguards. An option not available is Not Certified, because Not Certified means that the system would no longer be permitted to run on the computer. The second constraint was that the evaluation and certification process should be able to be accomplished with three personnel or less in 3Ø days or less. Before proceeding I should like to point out that in NASA the term used for the technical review of the application is evaluation rather than certification as it is used in the DoD. The certification step is the sign-off by a management official which is equivalent to the accreditation step in the DoD.

I would like to point out that one of the primary things that makes the evaluation and certification process work well in NASA is the computer security management structure within the NASA Centers. Each Center has a designated Center Computer Security Official; there is also a designated data processing installation Computer Security Official; and, for each sensitive application, there is a designated Application Computer Security Official. In those instances where an application runs on a stand-alone mini or micro, the same individual may be the DPI and the Application Computer Security Official. It is important to note that within NASA the Application Computer Security Official is the certifying official.

There are a limited number of sensitive applications in NASA. The total of 6Ø covers all nine Centers. All applications in the sensitive application inventory are in the administrative and business areas. One of the things I failed to mention earlier is that this approach is oriented to sensitive unclassifed applications in NASA and is not intended for use with classified applications.

The approach used by NASA was developed to address the need to perform an initial evaluation and certification of existing sensitive applications. The approach, since it focuses on the initial certification, is not bound by the independence issues in OMB Circular A-71 in conducting an evaluation or audit for the purpose of recertification. The approach relies heavily upon the knowledgeable users and developers. The NASA approach is tied very strongly to the system life cycle development process. The evaluation does not include a review of the data processing center, rather it looks at the application in terms of how it operates functionally and the data that is being processed by the application.

In the early stages of developing the approach it was recognized that not certifying an existing application was not an option available to us. In terms of objectives, we wanted a process that was practical and that could be accomplished by three persons in 3Ø days or less.

We referred back to the system development life cycle and initially focused on the security requirements and the technical specifications. We also realized that in the case of existing applications that had been around for some period of time, there probably were no documented security requirements or technical specifications. We did feel, however, that if nothing more was accomplished in the evaluation than getting the requirements and specifications on paper, we were about a thousand percent ahead of where we were when we started.

I would now like to spend some time in the area of evaluating the existing safeguards in applications. But before I do that, let me bring to your attention some portions of FIPS PUB 73 that we found useful in addressing the area of security requirements. FIPS 73 provides a six class taxonomy of systems that describes types of applications that have common security objectives. We used the taxonomy by first identifying the type of data processed by an application to determine the security objective or objectives. Using the security objective as a pointer, one can then look at the basic controls that satisfy the objective and use the description of the basic control as a statement of security requirements. If security require-

ments and specifications are already documented, going through the exercise using FIPS 73 can serve as a checklist to see if the requirements are complete.

Now on to the area of evaluating the current security safeguards in the application. In the development stages of the methodology we wanted a fairly reasonable and simple way to use the knowledge possessed by the users and developers in understanding the current security posture and weakness of the application. We went to an approach that had been written and published by Professor Brandt Allen of the University of Virginia at Charlottesville. Professor Allen refers to the technique as Threat Fraud Team Analysis. It was originally used in looking at a number of insurance and financial applications. We used the technique in doing a threat and vulnerability analysis with approximately ten users and developers. The first step was to relate a few war stories of how similar applications has been penetrated and exploited for personal gain. This is done to raise the participant's level of awareness and paranoia. The team was then asked how could you, singly or in collusion with another person, exploit the weaknesses in the system. It was quite amazing to see that in just a few minutes that people started coming up with several scenarios of how the current safeguards could be defeated due to inherent weaknesses or lack of controls in a number of areas.

As the scenarios began to unravel, we documented which module of functional area would be attacked, the vulnerabilities that permitted the attack to succeed, the likelihood of success and the amount of a one-time take. Next we asked the participants to identify what added measures could or should be taken to reduce the likelihood of the attack. Proposed safeguards were then categorized as critical, necessary and desirable.

For the system that was the subject of the test of the evaluation approach, it was recommended that the application be certified with qualification for 12 months pending the implementation of the critical controls. At the end of the 12 months it was also recommended that the application be reviewed again. In this test we identified some six scenarios and proposed about 12 to 15 controls of which six were considered critical. Two sessions were held. There are some advantages and disadvantages in using the threat scenario analysis and I think they are pointed out very well in Brandt Allen's article. The team members should be personnel in whom you have a great deal of trust. It should be realized that as a result of the threat scenario sessions you may have trained some potential penetrators who now have the knowledge to exploit the vulnerabilities identified in the evaluation exercise.

The advantages of the approach are: (1) the analysis points out where there may be problems in the system development process; and, (2) where controls are lacking or ineffective.

There are some additional cautions in using the threat team approach. Do not publish the fact that the threat sessions are being conducted and do not openly circulate the results of the sessions.

In summary, this methodology was designed to consider the NASA environment. It provides an evolutionary approach that addresses a complex problem and at the same time gives us an opportunity to learn and gain some experience in the area of certification.

Thank you for your attention.

REFERENCES

Allen, B., "Threat Teams: A Technique for the Detection and Prevention of Fraud in Automated and Manual Systems," *Computer Security Journal*, Spring 1981.

Fitzgerald, J., "Developing and Ranking Threat Scenerios," *EDPACS*, September 1978.

Garrison, H.F., Jr., and Simpson, G.A., *An Overview of ADP Risk Analysis*, MTR-79WØØ445, The MITRE Corporation, November 1979.

NASA, Appendix J, "Computer Resources Management," NHB 241Ø.1.

National Bureau of Standards, FIPS PUB-73, *Guideline for Automatic Data Processing Risk Analysis*, August 1, 1979.

OMB Circular A-71, Transmittal Memorandum No. 1, July 27, 1978.

Tompkins, F.G., *Security Planning for Computer Applications*, MTR-81W3Ø2, The MITRE Corporation, December 1981.

PANEL SESSION - AVAILABLE COMPUTER SECURITY PRODUCTS SATISFYING STATED REQUIREMENTS

Moderator - Mario Tinto

Chief, Products

DoD Computer Security Center

Panel Members:

Stan Kurzban - IBM,Inc.

Linda Vetter - SKK,Inc.

Terry Cureton - Control Data Corp.

Benson Margulies - Honeywell

Paul Cudney - System Development Corporation

Lester Fraim - Honeywell

INTRODUCTION

The purpose of this session is, as the title implies, to show how specific industry products address various requirements of the Trusted Computer System Evaluation Criteria. The first speaker will be Mr. Stan Kurzban of IBM, who will discuss RACF with regard to the requirement to enforce discretionary controls. The issue for discrestionary control is the need for the system to support need-to-know decisions or, in other words, the system must allow the file owner to control the sharing of his files and resources.

Stan Kurzban

Thank you, Mario. The first thing that I'd like to mention is that yesterday the question was asked, "Is MVS being evaluated?" You'll see from the top of the first foil that the evaluation that is being done is RACF running on MVS rather than MVS independently. In fact, MVS is serving as a basis for other products that are also being evaluated. So, I just wanted to clear that up first.

The requirement that I am going to be focusing on today is the requirement for discretionary access control with controlled sharing by named individuals or defined groups or both via access control lists. I'm going to be talking about how RACF meets that with profiles that correspond to the objects that are protected. RACF is the product that runs on the MVS operating system that's offered by IBM to address the matter of data security. It has underneath it the committment that the corporation has to the System Integrity of MVS and that addresses certain other requirements listed in the evaluation criteria and provides the basis for RACF and for the discretionary access control. It checks the authority of individuals called subjects in some contexts, to access individual things, called objects, in those same contexts. In addition, it provides other features that enhance the ability of management to manage what's going on in data processing, provides certain statistical objects and the opportunity to write installation specific exit routines to tailor the facility which is offered, to individual environments.

The objective of many of the features for discretionary access control is to reduce to a minimum the impact on individual users so that certain things happen automatically, that certain files, which we call "data sets," are protected automatically as the files are created, and that when the files cease to exist the descriptions of their protection also cease to exist automatically. The product consists of certain utilities and administrative commands as well as the actual code that performs the mediation.

The basis of access control is user verification. When users first enter into a series of interactions with a system, whether it's through batch or a series of interactions called a session the individuals have to identify themselves and demonstrate in some way that they are who they claim to be. Now, the traditional way to do that, of course, is with passwords. RACF also supports an operator identification card which is a card with magnetically readable information on it that verifies the individual's authority to access certain objects, to use terminals, to use certain applications, and membership in the group, a notion which is specifically called for in the evaluation and which is supported by RACF. Now, the individual resource managers in MVS call on RACF to perform the necessary mediation. So, for example, in the case of files, the data management part of the operating system calls on RACF and says, "Somebody has requested access to a particular object, should this request be granted or not?" and RACF makes that decision known to data management.

The authorities that are permitted are several: alter, control, read, write, update. The meanings of "alter" and "control" may not be obvious; they also may not be important in this context. "Read" and "write," I should think, are fairly obvious and so is "none."

The checks that are performed to determine whether or not access is to be granted are shown on the visual. If, in fact, the object is the personal property in some sense of the user who is making the request, then the request is granted and that seems kind of natural and obvious in a way. If the user has been specifically authorized to access the object, then the request is granted. If the group in which the user is acting in this series of interactions is authorized, that access is requested. It's not only what the object is, but what the level of access is - read, write, etc. that counts.

In addition, as Bill Murray mentioned on Tuesday, there is the notion of an "Operations" attribute so that operators are treated, as a class of people, in a way that is somewhat different from the way other people are treated; and there is a notion of universal access. For example, it may be that many system programs, the most basic systems programs, should be available to everybody. So there is one notion of universal access that enables everybody to use those programs. It is not sufficient for access that, in fact, the data set be associated with one particular group. That does not mean that everybody in the group can access the data set in any way that the individuals please.

Now, it is important to note that as suggested by the Evaluation Criteria, the type of access control that I'm referring to is one in which a user identifies him or herself and then all access control after that is automatic. The system "knows," in a very real sense of the word, who is authorized to do what. This contrasts with the notion in the previous session: the word "lockword" was used for this concept, the notion of passwords associated with resources. In a FIPS Draft Standard that's called a resource oriented password, if I remember correctly. There are several disadvantages to that scheme. That is the reason that we went to something like RACF although the scheme is in use. I'll mention a couple of its advantages.

If you have a password associated with one particular resource, then the password must be known by all the individuals who desire access to the resource. That means that if the password becomes known to somebody else, you don't know who lost it because there were a large number of people who knew the password and any one of them could've been responsible for the leak. If you wish to withdraw access from one individual but leave it with all the others, N minus 1 individuals, (this won't be a mathematical talk, but a few variables will creep in here) the other N minus 1 individuals must now be informed of the new password which will have to be used to deny access to the person from whom access has been withdrawn.

Users are authorized to use particular resources. They are also members of groups as suggested in the overview in the beginning and they have certain attributes associated with them. I've already talked about the "Operations" attribute.

There's also a notion of an attribute called "Special" that is administrative in nature, and also, the notion of "auditor." Another thing that Bill Murray alluded to on Tuesday when he talked about an individual who can observe a great many things, but do very few things. The notion of Auditor is treated special.

There are also special facilities. One of them is called ADSP. That stands for Automatic Data Set Protection which insures that all objects created by the subject are automatically protected. That shouldn't be a very strange notion. It is not appealing to all users of operating systems; I suspect that it has great appeal to the Defense Department.

Group access is a notion of access rights that are associated with membership in a group and "revoke" is listed as an attribute. That is, that there is a notion of revoking an individual's right to access anything in the system. That's the person who has just been discovered to be a spy, who has just been terminated, with or without extreme prejudice or whatever, who is no longer a member of the organization. There is also a notion of class authorization which particularizes authorization to types of objects, only volumes of magnetic tape or only terminals or what have you.

The groups are used for authorization to resources as has already been mentioned. They also afford a level of protection of control over direct access data sets and there will be more about that on the next visual. It also permits decentralized or centralized control of access control because individuals can be granted particular rights within groups, so that administrators can be assigned for specific groups rather than only one system administrator who must do everything.

The administrative commands process the RACF profile. This is a very important notion, the notion of profiles. Here is where the math comes back. Consider a set of S subjects, those are the users, and O objects. Then, obviously, all the permutations of access even if we assume only one type access is S times O. We have to ask the question, "Can each subject access each object and then if so, how read, write, modify, etc?" Now, in fact, for most subjects, that is, if we pick one subject at random and one object at random, the chances are very good that the type of access which is permitted is "none." That says that most entries in the matrix describing the authorization are zero. The result is what is sometimes called the sparse matrix, and if you want to store the sparse matrix, it can be efficient to store only nonzero entries.

You have two obvious choices in storing the non-zero entries and one unobvious choice. The obvious choices are: You store only the rows and indicate which columns have non-zero entries, or the opposite. So you can have a list for each object and say which subjects are authorized to access the object, or you could have a list for each subject and use the notion of capabilities also known as tickets, the rights to authorize particular objects.

Another scheme, of course, is the notion of just storing rules. The rules enable you to figure out which subjects can access which objects and the rules can be driven off the names of the subjects and the objects. What we're talking about in RACF is access lists, as called out specifically by the Criteria. That is to say, associated with each object is a list of all the individuals who are authorized to access the object and in what ways that access is authorized. Now, in addition, because there are profiles, one can store in the profiles information about the object. One can store notions of ownership, statistics about the object as to how frequently it is used or how big it is or to whom it should

be charged or when it should automatically go away. These are all possibilities, not necessarily realizations. One can store a notion of level.

Now that's an important notion that people have been heard to talk about earlier in the week, too, and I'd like to say something about that. Certainly, the notion of level can include classical levels, TOP SECRET, SECRET, CONFIDENTIAL, and cleared information, but the notion of level in terms of RACF totally, is not carried through as it is in the classification scheme used by the Department of Defense. There is no notion of a level associated with an individual's subject such that that subject can access all things or some subset of the things that are at the SECRET level or the TOP SECRET level. That notion of clearance for an individual or association of a level for an individual is one example of what might be called the pervasive policy of levels. When you mark some documents CONFIDENTIAL and there are other documents that you do not mark CONFIDENTIAL that is not a pervasive policy.

The notion of APF authorization is one that exists in MVS. It is a notion that separates privileged things from unprivileged things, in particular privileged programs from unprivileged programs and the commands that are part of RACF operate with APF authorization in most cases. There is enforcement of the authorization scheme, mediation, and error recovery for the commands. That, I hope, goes without saying, but is another item. The commands in particular are the ones shown, the basic commands for discretionary access control. One can add, alter, list, or delete a profile. Sometimes verbs don't match up with nouns or objects very well, but in this case I think that it will work: the nouns in this case are user, group, data set, and other resources.

We're getting close to a summary. We have automatic data set protection to ensure that newly created things are protected as soon as they are created. They don't lie around unprotected until somebody gets around to it. We have data set profiles and we have the very important notion of modeling and this contributes to the notion of making it possible for the users to be as unaware as possible of security schemes, because when you have users who are very conscious of what is being done for security, you have natural antagonists of your security scheme. So, modeling is one of the mechanisms used to make sure that things can be done fairly easily as long as the things that are being done are things like other things that have been done in the past. Set up a model and then if nobody takes extraordinary action the protection afforded the new data set is just like the protection afforded the data set that was created previously and the system knows about that and does that without any visibility to the user. There are other notions of default which permit this ease of use. For example, universal access which I alluded to earlier, the notion of an owner of a data set, levels which I mentioned earlier, and the notion of auditing. Auditing is controlled by the auditor and associated with classes or groups of things all at once and does not have to be specified individually for individual things.

In summary then, we try to provide accountability to the user as called for by the Criteria, we try to provide transparency for the user which I think is not as prominent in the Criteria, but is obviously desirable, and we try to provide installation control, and you won't find anybody to argue against that objective.

So, we are back to where we started, talking about how RACF on MVS meets discretionary access control and the other requirements of the Criteria. What I've done on this and the next two foils is to identify particular requirements and to give some indication how RACF meets them, so you can understand that RACF, while it has the discretionary access control I've described, is not all we're talking about. When RACF runs on MVS, it provides for: identification and authentication; protection of authentication data; and associating a unique identity with all audible actions. That's done by RACF's checking of passwords and the operator ID cards I mentioned. Auditing is very important and I've taken some words out of the Criteria to describe that requirement. RACF records what has been done via the System Management Facility of MVS (SMF).

The requirement for System Integrity (and I'd like to make a special point on this notion), the very System Integrity statements that are put out by IBM to express our committment to the integrity of MVS and various program products including RACF call out RACF specifically. That committment states specifically that the corporation's committment to System Integrity (what we call System integrity; you might call it operating system security, depending on how you define various terms), the notion that unauthorized things shouldn't be possible, calls out specifically RACF and the objects that are protected by the use of RACF, and that RACF itself, if installed, would not compromise that protection.

Another requirement is that it should not be possible to view something in main storage just because it wasn't erased before new storage was allocated. That's a property that MVS and even at least one of its ancestor systems, going back about twelve to fifteen years, had. That is a B3 category. In addition, requirements having to do with documentation and resource encapsulation are met at the C2 level. This is not meant to be an exhaustive list of what RACF on MVS satisfies, but I understand that such an exhaustive list will be forthcoming soon. So, when you get that evaluation, you will have the full story. Thank you.

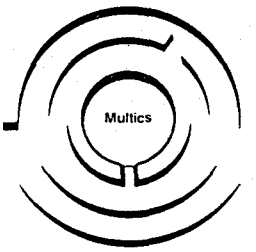
The next speaker is Benson Margulies from Honeywell's Cambridge Information Sciences Laboratory (CISL), who will discuss the MULTICS system and mandatory access controls. The mandatory access control requirement is for the system to be able to apply a pre-defined set of access rules based upon the attributes of both users and data/resources. In the DoD context, mandatory access controls speaks to the requirement to support the classification and compartmentation structure, also known as the lattice model.

Benson Margulies

NO TEXT AVAILABLE. Following are the slides used by Mr. Margulies in his presentation.

Multics
Access Isolation Mechanism

Compartmentalized Access Control for a General Purpose Timesharing System



Slide 1

Multics AIM

- Two approaches to access control: solving the Trojan Horse problem.
- A brief introduction to Multics files, address spaces, and hardware access control.
- Multics Access Isolation Mechanism (AIM)

Slide 2

Multics AIM

Two approaches to access control

- Discretionary Control
  - — owner specifies access rights of users to objects.
- Trojan Horse problem
  - — programs running in trusted process can misuse discretionary control to leak information.
- Nondiscretionary Control
  - — imposes structure of security classifications that further restrict access, and are not controlled by owner.

Slide 3

Multics AIM

Discretionary and Nondiscretionary Control

Discretionary Control, an example

Associate a list of users and permissions with each object:

Margulies	read
Tinto	read, write
*	read

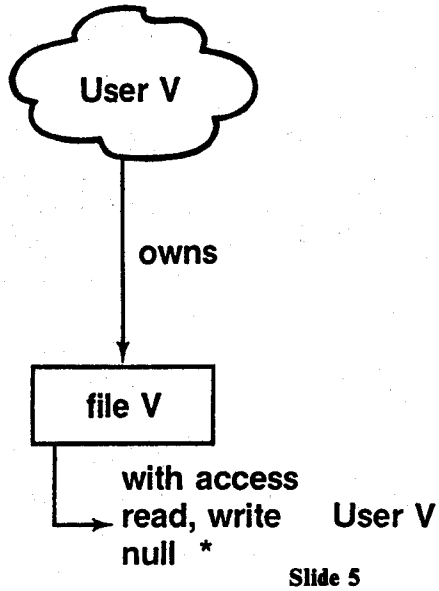
Search the list to determine effective access.

	Margulies	gets	read
	Organick	gets	read
but	Tinto	gets	write, read

Slide 4

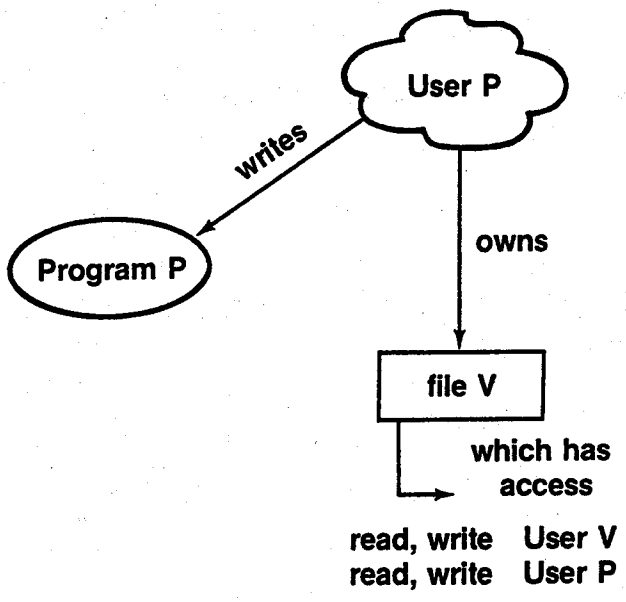
Multics AIM

Trojan Horses



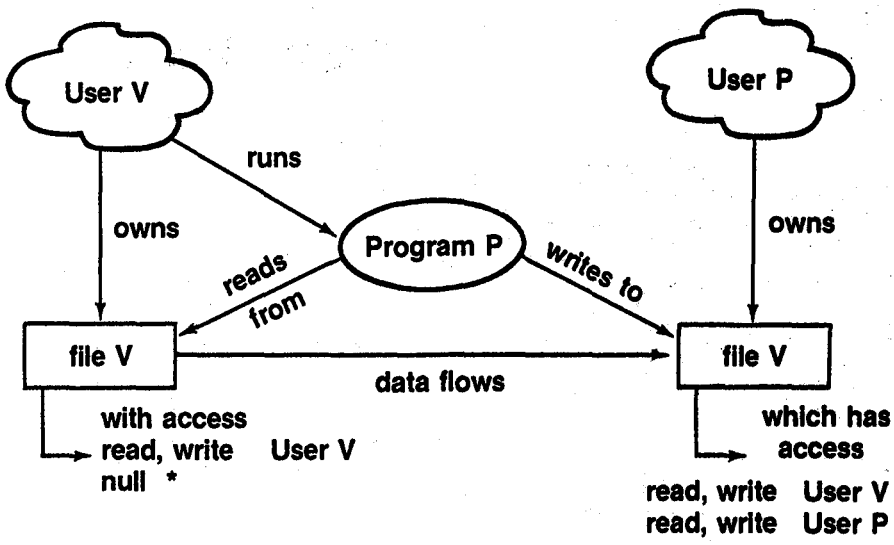
Slide 5

Multics AIM



Slide 6

Multics AIM



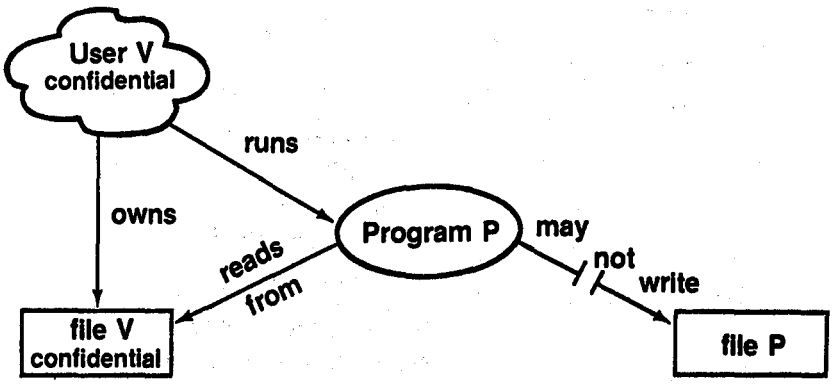
Program P is a "Trojan Horse"

Slide 7

Multics AIM

Nondiscretionary Control

- Classify information permanently
- Enforce restrictions based on class



Slide 8

Multics AIM

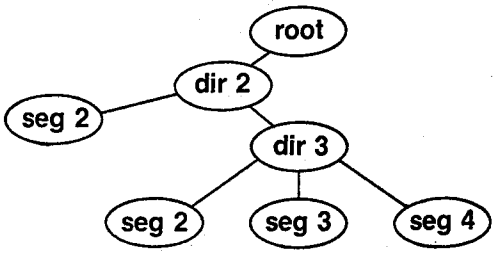
- files and directories
- users and processes
- address spaces and virtual memory
- hardware access control

Slide 9

Multics AIM

Files and Directories

- files — called "segments"
  primitive access — read versus write
- directories — tree structured hierarchy of segments & directories



- primitive access — status versus modify
- discretionary Access Control Lists

Slide 10

Multics AIM
Users Processes

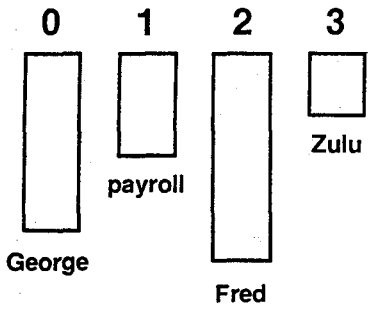
- Users — have names & passwords
- Processes — users' agents in system
  - — run programs ··· "control point"
  - — reference programs and data ··· "address space"

User supplies name and password, and system creates process.

Slide 11

Multics AIM
Address Space & Virtual Memory

- Segmented address space



  - — segments contain programs and data
  - — segments are variable in size
  - — hardware addresses are segment numbered and offset.

- Virtual Memory

Segments in address space = = segments in file system!

Segments are added and removed dynamically.

Slide 12

Multics AIM
Users Processes

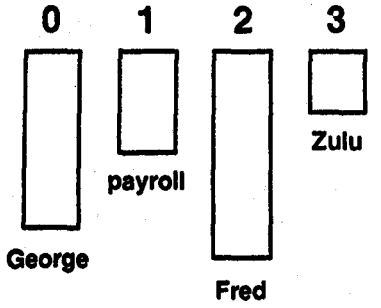
- Users — have names & passwords
- Processes — users' agents in system
  - — run programs ··· "control point"
  - — reference programs and data ··· "address space"

User supplies name and password, and system creates process.

Slide 13

Multics AIM
Address Space & Virtual Memory

- Segmented address space



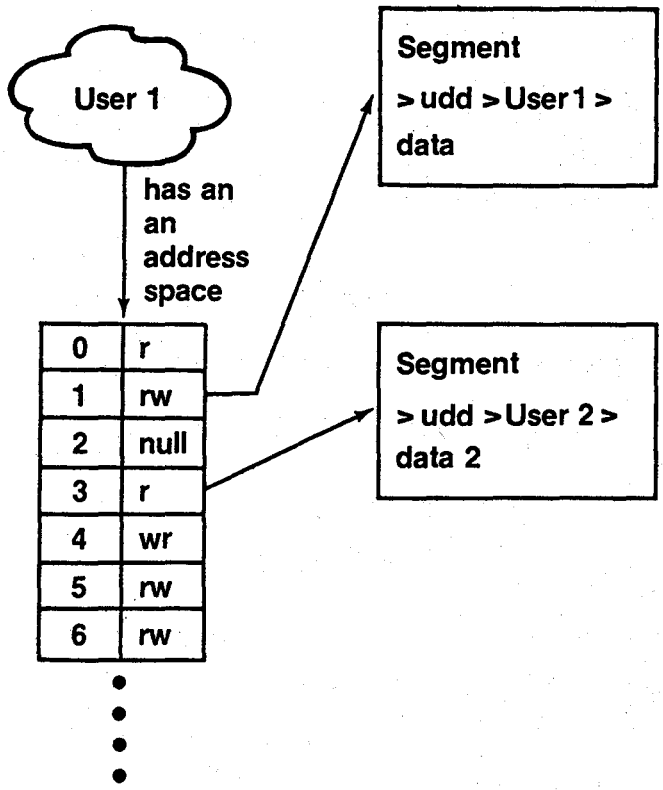
  - — segments contain programs and data
  - — segments are variable in size
  - — hardware addresses are segment numbered and offset.

- Virtual Memory

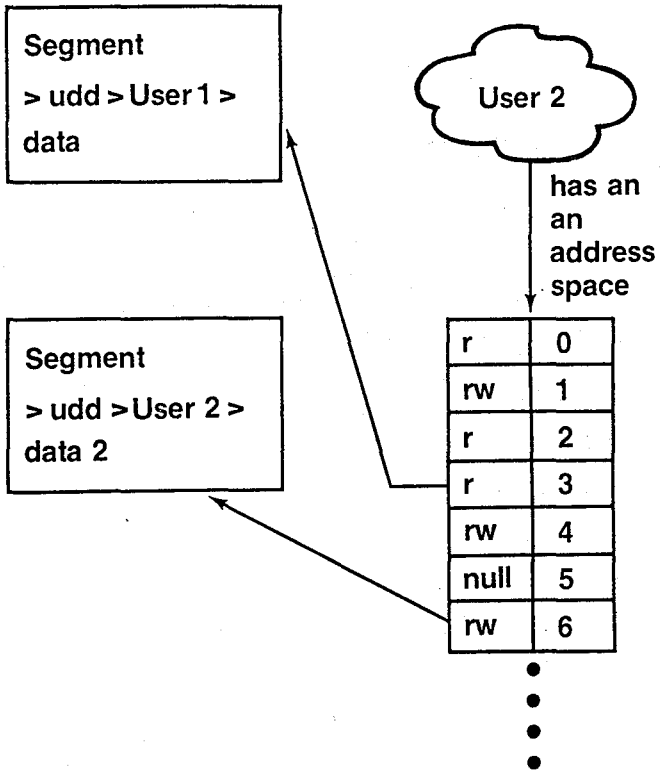
Segments in address space = = segments in file system!

Segments are added and removed dynamically.

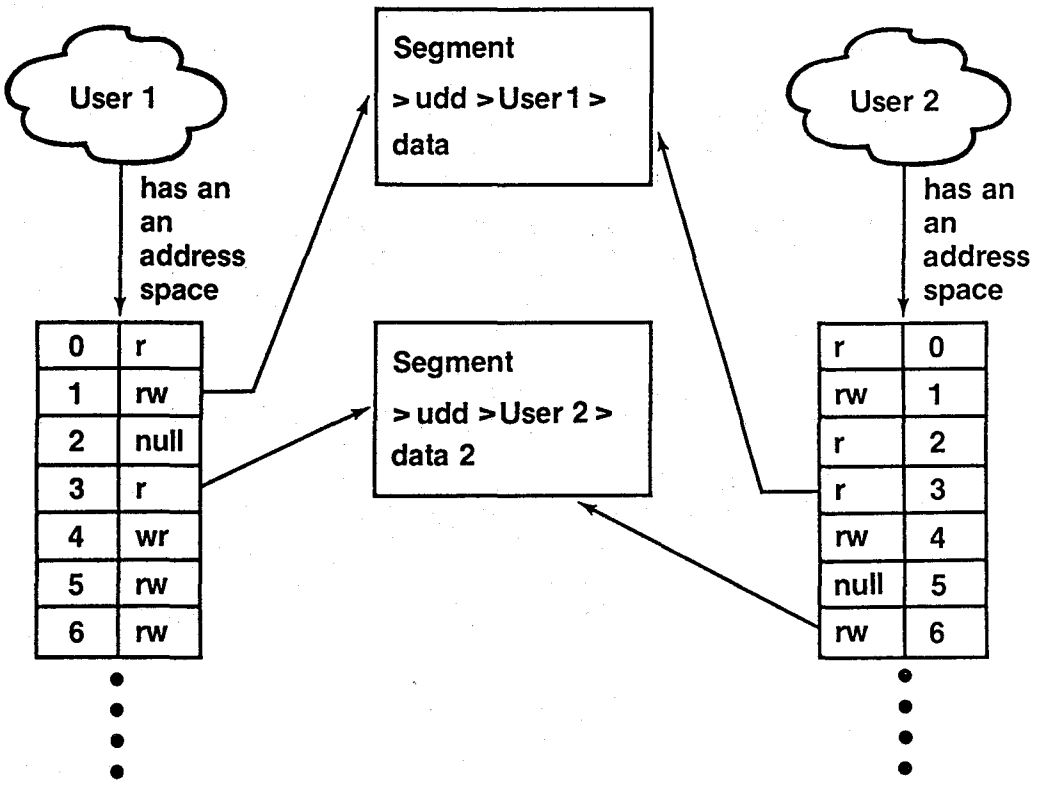
Slide 14


User 1
has an
an
address
space
0
r
1
rw
2
null
3
r
4
wr
5
rw
6
rw
Segment
> udd > User 1 >
data
Segment
> udd > User 2 >
data 2

Slide 15


Segment
> udd > User 1 >
data
User 2
has an
an
address
space
Segment
> udd > User 2 >
data 2
r 0
rw 1
r 2
r 3
rw 4
null 5
rw 6

Slide 16


User 1
has an
an
address
space
0 r
1 rw
2 null
3 r
4 wr
5 rw
6 rw
Segment
> udd > User 1 >
data
Segment
> udd > User 2 >
data 2
User 2
has an
an
address
space
r 0
rw 1
r 2
r 3
rw 4
null 5
rw 6

Slide 17

Multics AIM
Access Isolation

- Classifies all information on a Multics System.
- Run or not at Site Option
- 8 years old
- Used at AFDSC, Oakland

Slide 18

Multics AIM

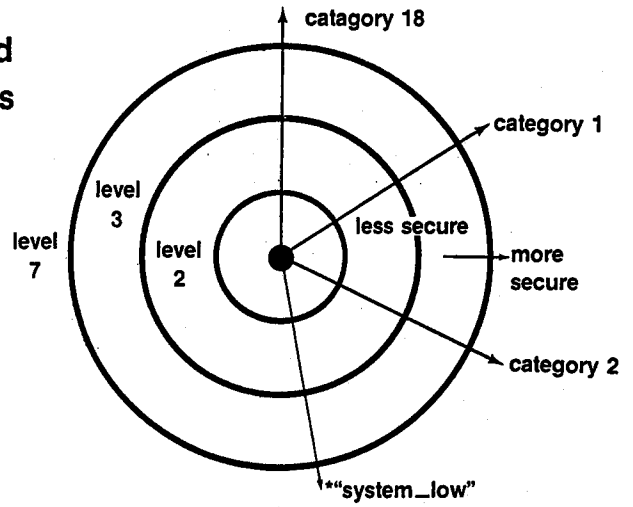
AIM

- The Classification System
- Basic Rules
- Marking of Objects and Processes
- Rules for Objects and Processes
- Security Audit Trails

Slide 19

Multics AIM
Classification System

Levels and Categories



— each object is in <u>one</u> level and <u>one or more</u> categories

Slide 20

Multics AIM

Classification System

Example:

categories: Personnel, Development, Dirty Tricks

levels: sensitive, very sensitive

Typical classifications:

system – low

personnel, sensitive

personnel, very sensitive

personnel, development, sensitive

Slide 21

Multics AIM

Rules

— no information may flow from a higher to a lower level.

— no information may flow across category boundaries.

⟷ no category mark may be removed from information.

Slide 22

Multics AIM
Marking of Objects

- Segments & Directories — access class
- I/O Devices — range of potential access classes, — current class
- I/O volumes — access class

Marking of Processes

- Processes have authorization — level & categories

Slide 23

Multics AIM
Rules for Processes and Objects

1. no process may READ unless
    a. its level is >= the object's level
    b. its categories contain the object's categories
2. no process may WRITE unless
    a. its level is = the object's level
    b. its categories are identical to the object's
3. no process may send a wakeup to another process unless
    a. sender level is <= receiver
    b. sender categories are identical to receiver.

Slide 24

Multics AIM
Security Audit Trails

Log all failures to pass access control

— at addition to address space time

— at certain hardware faults

— at directory accesses.

Slide 25

Multics AIM
Concluding Thoughts

- AIM provides compartmentalized access
- It requires that people
  - — physically secure facilities
  - — enter information appropriately
- Does not degrade system performance significantly.

Slide 26

At an earlier session we heard Mr. Barry Schrager, President of SKK,Inc. briefly discuss their product, ACF2, and its history. Our next speaker is Linda Vetter, also of SKK, who will present ACF2 in more detail, focusing on the Criteria requirement for authentication and auditing. Here the issue is individual accountability, or the need for the system to be able to unambiguously ascribe each event to the actions of a specific individual.

Linda Vetter

There is going to be a little bit of redudancy between some of the things I'm going to talk about and some of the things that Stan already talked about in relationship to RACF, but I'm going to try to put a little different flavor in it and come up with a few new points.

I'm only going to focus on a couple aspects of ACF2. Basically, identifying users and auditability. So this is the only foil I have that gives you an overview of ACF2 as a product. It is a very brief overview. Basically, ACF2 has two major functions: controlling access to the system itself (in other words, who can log on or sign on to your system or run a batch program on your system) and secondly, once the person is on the system, what resources associated with the system they have access to. Can they issue a particular transaction on your data base system? Can they access a particular file? ACF2 operates predominately by adding extensions to IBM operating systems, on IBM and IBM-compatible equipment. Stan talked a little bit about MVS as base and that's what has been evaluated at the Computer Security Center - ACF2 on an MVS base. ACF2 also runs on VSI and VM operating systems. Some of the design philosophy behind ACF2 obviously includes the individual accountability and the auditing information which I will talk about in a little more detail in a second, but one of our important underlying design philosophies is protection by default. This means that ACF2 is designed to control the sharing of data. If you are not preauthorized, if it has not been stated that you are a legitimate user of the system or that you have access to a given resource in a fashion that you have requested it, the access will be denied. The default is always that access is denied, so that when controlling or sharing of data, you don't have to necessarily know all the levels of data. You need only to identify in which cases you want to share it; the default is that unauthorized access will be prevented.

Data on foil number 4 is the evaluation information or criteria that we had originally to work with. Obviously, this has now been at least superseded by the orange book. Hopefully, the content is still basically the same. In the identification authentication area, the Criteria states that the TCB must require users to identify themselves before they perform any other action that the TCB is supposed to be controlling. TCB also has to provide enforced and protected authentication. It has to protect that authentication data. In other words, if you are using something like passwords, you don't want the passwords lying around in the system so that someone else can get a hold of that data. It must enforce individual accountability. All the actions must be tied to individual users throughout the life of their use of the system, plus the ability to then associate that identity with everything they do, in an auditable fashion.

Some of the ways that ACF2 handles this area is that it validates every request to get on to the system. As I said earlier, it can be a time sharing "logon" operation or the submission of a batch job, or signing on to a data base transaction-type system. It could be an operator started task. In any event, ACF2 is going to validate that the ID to be associated with that job: 1) has been predefined to ACF2 as a legitimate user of that system, and 2) has been preauthorized to use the system in the way in which he is attempting to access it. One of the things that we are going to look at is the time of day or the day of the week that he is getting on the system. So you can define that the user can only access the system from Monday through Friday from 9 to 5, and if he's trying to come on the system at a different point in time, you can deny access. We also look at the actual physical device he's using to get on the system or the input source that's being used. You may predefine that your payroll clerks can only get on the system using the terminal in the accounting department, and if a payroll clerk tries to sign on the system from the warehouse, you know right away that you have a potential problem and, of course, ACF2 would deny that access. You can also specify certain sub-systems. For example, you can allow your programmers to only sign on to your test version of your data base system if you have CICS, IMS, or a similar data base type system. You can define for each version or each region that is operating that kind of system if the user is authorized to get on to that system or not. You could specify that the programmers are able to use the test system and the production people can use the production system.

Last but not least, is the user himself. The user's identification is verified predominantly by the password, although there are other methods. Before I talk about ACF2 and passwords more specifically, I'm going to digress a moment to something that Stan has already opened up for me and am just going to support what he said. There are different kinds of passwords. Historically I think passwords have gotten a little more bad press than they deserve because in many of the older systems and even in new systems, there hasn't been a good differentiation between what I would call a resource password as opposed to an individual user password. A resource password is tied to a resource. An example would be a data set password, where anyone that needs to have access to that data set shares the knowledge of that password and that password says, yes you can have that data set. Similarly, it could be an access code or it could be a password associated with a transaction in a data base system. Now, a shared password is like a shared secret. Obviously, it isn't as safe or unique or as protected as you would like it to be. Stan mentioned that if something happens, you don't know who to hold accountable. If ten people know that password and the eleventh person finds out, it's very difficult for you, even if you have a good suspicion who might have compromised it, to actually hold that person accountable because he can always say that "nine other people knew it and you can't tell me that it had to be me that gave it away." In addition, you have administrative problems, where if you change the password, you have to make sure you notify everyone. If one person leaves that may know it, you have to be sure you know about that and you do get the password changed. So that there are a lot of problems associated with shared

passwords. In the past they have been used to quite an extent and obviously, misused.

In the area of an individual password, the object is to tie that password to one individual and one individual only. There should only be one person, not only just one who *does* know that password, but only one person who *can* know that password. It is very important, if you are going to hold a person accountable (and we are talking about individual accountability here) that you can also guarantee that person that if he does not give his password away, if he is not careless with it (doesn't write it down, doesn't pick one that anybody would be likely to guess as being his password, etc.), then you can guarantee that if he is careful with it, the system will also be careful with it. The system will not expose it or give it away for him. Then you can go down and enforce your auditing tools, all the rest of your policies, and if something illegal or inappropriate does occur, you can go back to him and say this access has been associated with your ID and you're the only one who can use your ID, you're the only one that knows the password with it, you can be held accountable. If he says, I really didn't do it, you can then demonstrate that the system did not give that password up or that there is no one else who could know it from the system, then he is still held accountable. He was careless with it, typed it in with someone looking over his shoulder, or in some other fashion loaned it out, whatever. So it is very important that if you are going to use individual accountability, you can rely on the password to some extent. I am the last person to say that it is a foolproof system or that it is the only thing to use, because the more additional hardware or other kind of control devices you can add to the password you're going to have a better level of security. But individual passwords are certainly a very, very effective, a very useful method, if implemented in the correct way - which does include keeping it at the *individual* level.

Going back to ACF2, and talking about protecting that password, there are a couple of different things. First of all, in controlling passwords within ACF2, there are a number of options that you set up on an installation level on how you want to control the protection or use of passwords - keeping them secure. Obviously, you can identify who has authority to change the password. I personally believe it should always be the user who changes his password and he should not be assigned passwords centrally. However, we do provide both options because some sites feel very strongly the other way. The reason I believe it should be the individual only is, going back to the individual accountability point, if the individual is the one who creates it, changes it, and maintains it, not only is he more likely to remember it and less likely to write it down, but again you can enforce the fact that he is the only one who knows it. It was not assigned to him, whether it was put in a sealed envelope or not, there was no other person or program who knows what that password was or knows the algorithm of how it was created.

How often passwords are changed is another option of control in ACF2. For example, you can specify on an individual basis how many days maximum a person is allowed to go without changing that password. For a security officer you might say he has to change it once a week, while for a clerk in some operation you might say once every ninety days is adequate. You can also specify a minimum number of days. He cannot change it daily, for example, if you have some reason for that. And you can specify the minimum number of characters that has to be used in a person's password. You can set up your installation to say users must use at least eight characters, for example, in their passwords. He cannot use initials or short words that are easier to guess. You can also specify certain criteria to be imposed at you installation. I'm not fully in favor of this either because I think the more formatting restrictions you put on passwords, the easier you make them for somebody to guess. For example, if I knew they have to be two numerics followed by three alphabetic characters, that really limits the number of combinations that I would have to try to test if I was going to try to guess somebody's password. So, if you are going to use ACF2 options to enforce certain standards, be careful in how you do that and that you recognize those other trade-offs. You can also specify if a person can change his password when he gets on the system, or if he has options to change it at other times. You can also enforce that users change their passwords at any point in time that you feel it may have been compromised. You can force all users to change their passwords on the same day if you have some reason to be concerned about some sort of compromise.

The handling of invalid passwords within ACF2 includes a number of actions. Obviously, one of the most direct ones is that any effort or attempt to enter a password which is considered invalid (unmatched) is always logged. It always shows up on the audit trails reports. It can immediately show up as an on-line message at the security console or optionally at the operator's console, as well as on the batch reports.

In addition, the batch report will identify the ID that was being used, the actual physical device the person was at, the time he tried to do it, and, if a program or anything else was associated with it, what these other conditions were. These attempts are also counted, and the installation has the option of specifying how many tries they are going to let their users have before they are going to lock them off the system completely. In other words, you can specify that in a dial-up case that you will allow them two tries then disconnect the line, and in a global case, you will allow them to have maybe three or four tries and then will suspend that ID. That ID is then suspended by ACF2 indefinitely. He is not allowed to get back on the system until a security officer, who is authorized to control that particular user or be responsible for that user, re-authenticates the user and says he can get back onto the system. In that case, what normally happens is that the security officer will assign a new password for the user and the user will be forced to change that password immediately the first time he logs back onto the system. It is part of the structure of ACF2 that if someone besides the user changes the password and you are controlling it on a user basis, then the user has to change it the next time he gets on the system so that the minimum amount of time will lapse that you have more than one person that knows that password.

Last but not least, consider protecting the password internally. Again, I said that you can't hold that user accountable for not giving it away if the system is going to give it away. There are a number of things ACF2 tries to do to protect the password internally. One thing is we will not accept the password unless the user gives it to us in a

secured fashion, so to speak. For example, on a screen we will prompt for the password in a display protected area so that as he types it in it does not physically display. If it is on a hard copy terminal, such as old TI7ØØ or whatever he's got, we will create a darkened area (an x'd out area, a number of overprint characters first) and then he would type over that area. We try to keep it from being visible as the person is actually typing it in. Now the first thing ACF2 does with the password when it gets it, regardless of what kind of source it is coming from, is to encrypt it. We one-way encrypt, which is very significant. The one-way encrypted version is the format that we use to actually store it on our data base (where we are keeping track of ID's and passwords) so that if anybody ever was able to get access to that data base, they could never see anything but these encrypted passwords. They are one-way encrypted, which by definition means that they cannot be decrypted. So if you're a user, you forget your password, you go to the highest level security officer in your system and ask him to help you to get back on the system, he cannot even tell you what your password is. There is no way that he can decrypt it or read it, or tell you what it was. He can reestablish you as a user. He can temporarily assign you a new password for your one access to get back on and change it, but he can't tell you what your old one was. Now our one-way encryption algorithm is actually a double encryption. The first phase is through our own unique encryption algorithm and we run it through a second phase which is a modified (extended) version of the Data Encryption Standard (DES). So, it actually goes through a double encryption, but it is all one-way encrypted. It cannot be decrypted. We then provide some additional options. For example, if you are in a network environment where you are transmitting jobs back and forth, we will encrypt some of that data including the password information as we transmit it across the network nodes. That does require that you have ACF2 on both systems so that when it gets to the other system, they can recognize what they just got passed, but it does give you that option when you are in that environment.

Last but not least, you have facilities in ACF2 to model a new user after a previously established user. You can say, I've just added clerk number 1Ø in this department. I would like to create their identification to be exactly like clerk number 9 with all similar privileges. You can do that with a simple command in ACF2, however, we know that you don't want to copy over things like the other user's password. So there is no way to ever move that field or otherwise display it even in its encrypted form. It is not kept around in plain text in the control blocks while jobs are processing or anything like that. The first thing we do when it comes in is encrypt it and then we carry it around and use it in the encrypted format from then forward.

Auditing - The Criteria at the higher levels states that to meet the auditing criteria the TCB should create, maintain, and protect records of accessors. That these records have to include the identification of the user, the object they were accessing, the kind of access that was taking place, and the time and date that it occurred. There should be options provided to selectively look at these reports and there should be documentation included representing what the records indicate and also, obviously, how to use the reports. I'm not going to talk in detail about some of the areas like documentation, but with the package of manuals that is delivered with ACF2 there is a specific manual called the Auditor's Guide. So if you are an auditor at an ACF2 site you had better read that manual. It gives you all these neat little clues how to figure out what the site did wrong in implementation, if anything. There are also training classes including auditor sessions and presentations at our user conferences.

Some of the monitoring tools that we provide within ACF2 let you selectively trace records or log records on different criteria. Obviously, one option would be to turn a trace on a given user. If you want to know everything that the user is doing on the system, you turn Trace on in his ID record and ACF2 will create a record and produce a separate report by that user chronologically on everything that he has done. A user can normally list his own logon ID record, his own ACF2 control record, to see what his default values are and things of that nature, but when a user lists his own record, ACF2 selectively only displays certain fields. One of the fields we will not display is if he is being traced. We won't tell him that. The auditor can see that because we let the auditor see most fields, but we don't tell the user.

A subset of Trace is the ability to turn on what we call TSO Trace for TSO (time sharing option) usage. You can ask ACF2 to specifically create a record and a report indicating every command a person issued while on TSO, including certain buffer information that indicates not only that he was editing a file, etc.

Monitor is another option that you can turn on for a user. Usually these are used somewhat in sequence - you might trace what he does if you are suspicious and then if you are really suspicious and want to catch him redhanded you use monitor. Monitor generates a message immediately the next time (or every time) he signs on the system or submits a job. The message is displayed on the security or operator console and says that the monitored person just got on the system and from what device, from what location. You can go out there and grab him at that point.

Log shift is another option. I mentioned earlier that you could specify the time of day or the day of the week that the user is authorized on the system. Log shift is really more of a privilege than an auditing tool, although it is both. If the user has the log shift privilege, it says he has the authority to get on the system outside of his normally specified shifts. So that if you do have a programmer who is supposed to be there 9 to 5 Monday through Friday, but he does get called in ocassionally during the middle of the night and you want him to be able to get on the system, you give him this privilege. It allows him on the system outside of his normal shift, but then it will log that he got on the system at that time so that, at least, you have an after-the-fact indication and you can go in the next morning and verify what was taking place or why.

You also have options in ACF2 to log any use of given programs. If you had any specific programs that you wanted to track, the zap utility programs for example, you put them on this list and whenever anybody uses it, whether they are authorized or not, that information is logged. You can also log all accesses to specific resources, such as certain data set names, or certain transactions, and produce separate reports for that. Then there are certain functions,

like the use of certain commands or the use of what's called bypass label processing for tape data sets. You can log any occurrence of that usage. Obviously, it wouldn't be very complete if we logged all this and didn't ever let you look at it, so ACF2 also provides a whole series of report generators to sort and edit and display all this information. The critical ones for this discussion is the third group on the foil - violations and the events. Any attempted violation denied by ACF2 is always logged, whether it is an invalid password or an attempt to access data under some condition you are not authorized. For example, you may be authorized only to read it. If you try to write it, we would deny the request and log it. You can be authorized to read it, if you are running a particular program, but denied reading it running any other program or using any other input device, and in those events, we would always deny it and log it. In addition to that, you can request loggings of any specific events which we talked about previously requesting traces or logs.

The last area for the auditor is we also provide certain on-line commands and privileges just for the person auditing the system. Now this could also be used by a security officer or a manager, depending upon who you give the privileges to. The first one is the one that is most unique. The others are just the commands to list or display users and rules and information. The first one, the show commands, show you the status of your entire operating system with ACF2 installed to ensure that everything is there the way you thought it was implemented. For example, you have the option of writing local exits. You can write some local exit code to do some special testing or special privilege granting or restrictions, however you wish. Any exit you have running in your ACF2 system will display with the show commands. The auditors do not have to go to the system programmer to ask him if he has got any code in the system because he may or may not get the complete answer, but the auditor (or the security officer) on his own can use these commands to identify any exits that are in place, where they are, the name of the module, etc. It will also show what options the site is using, the minimum number of password characters being enforced, anything of that nature. So that the auditor, without having to rely on the system programmer, can always identify how the system is set up and how it is being operated.

This is not a sales pitch. I represent SKK which developed ACF2. We support it, we do the enhancements, documentation, and everything else. It is marketed in the United States and Canada by Cambridge Systems Group. So they are the sales people, not us. If you do want any other information, please feel free to contact one of us and we will be glad to help you with any other questions that you may have.

IDENTIFICATION AND AUTHENTICATION

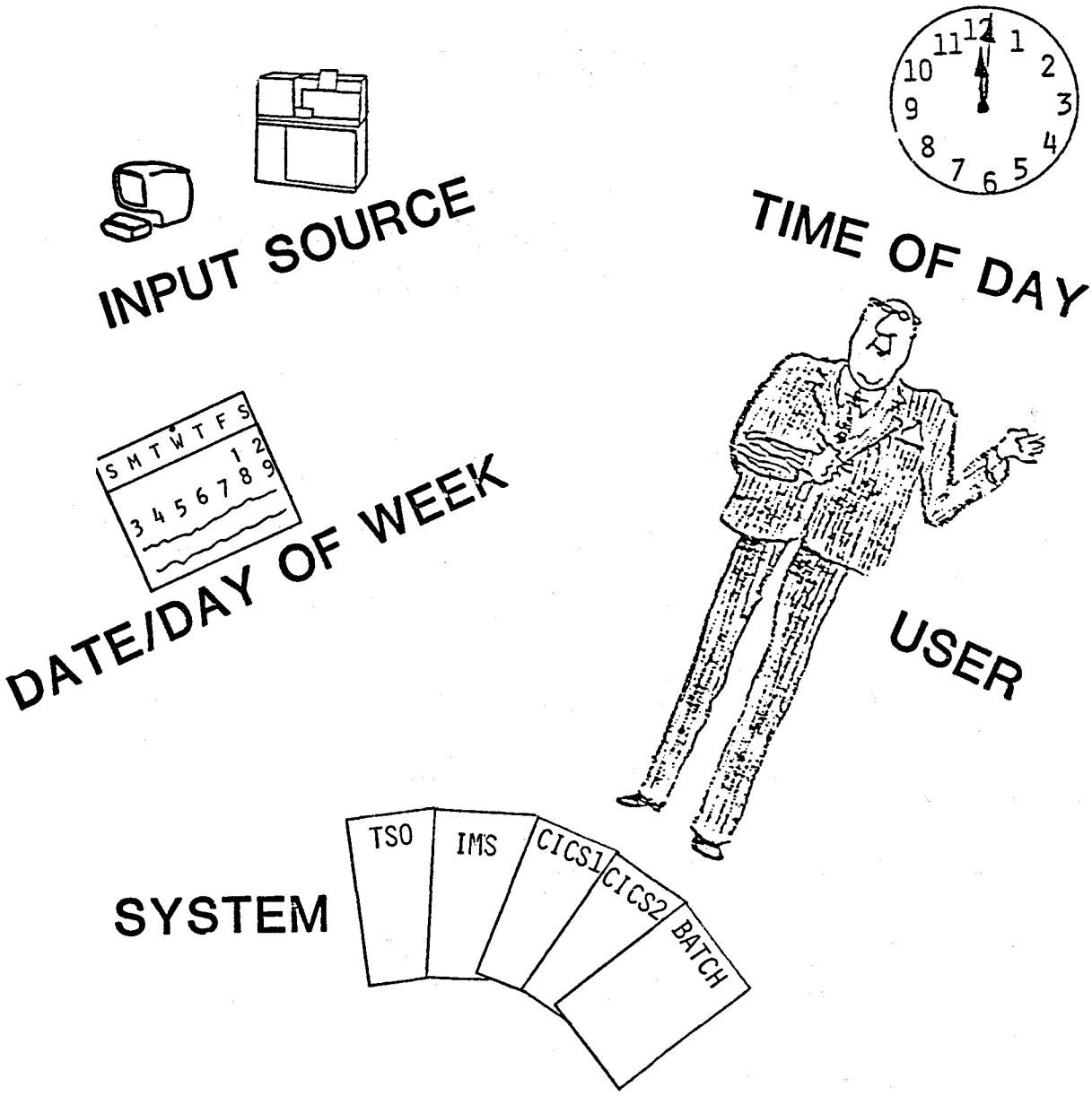
The TCB must

- require users to identify selves before performing any other actions.
- provide enforced and protected authentication of identity.
- protect authentication data.
- enforce individual accountability.
- provide the capability to associate this identity with all auditable actions.

Slide 1

SYSTEM ACCESS CONTROLS

Validate each request for correct --



Slide 2

PROTECTING THE PASSWORD INTERNALLY

- ☆ Optionally accepted on input only when prompted for (x-out mask or display prohibited area used).
- ☆ Never displayed by ACF2 commands or reports (encrypted or plain text).
- ☆ Immediately one-way encrypted on input, not kept in clear text.
- ☆ Stored on ACF2 control database in one-way encrypted format only.
- ☆ Network options to protect transmitted jobs' passwords.
- ☆ Cannot be copied from one user record (e.g., prototype or model) to another.

Slide 3

AUDIT

The TCB must

- create, maintain, and protect record of accesses.
- include indentity of user, object, type of access, and time.
- provide options for selective audit by user.
- include documentation on detailed audit record structure.

Slide 4

MONITORING TOOLS AVAILABLE

BY USER:

TRACE
TSO-TRC
MONITOR
LOGSHIFT

BY PROGRAM:

LOGPGM Program Name Traces

BY RESOURCE:

LOG Access Rules

BY FUNCTION:

Log BLP Usage
Log TSO Command Records

Slide 5

ACF2 STANDARD REPORT GENERATORS

Preprocessing/Statistics:

ACFRPTPP - PRE-PROCESSOR AND STATISTICS

Control Data Base Update Logs:

ACFRPTEL - INFORMATION STORAGE MODIFICATION LOG
ACFRPTLL - LOGONID MODIFICATION LOG
ACFRPTRL - RULE MODIFICATION LOG

Violation Attempts and Event Traces:

ACFRPTCR - TSO COMMAND TRACE
ACFRPTDS - DATASET/PROGRAM EVENT LOG
ACFRPTJL - RESTRICTED LOGONID JOB LOG
ACFRPTPW - INVALID SYSTEM ACCESS LOG
ACFRPTRV - GENERALIZED RESOURCE EVENT LOG

Additional Useful Utilities:

ACFRPTIX - DATASET INDEX REPORT
ACFRPTRX - CROSS REFERENCE (USER/RESOURCE) REPORT
ACFRPTSL - SELECTED LOGONID LISTINGS
ACFRPTXR - CROSS REFERENCE (RESOURCE/USER) REPORT

Slide 6

The next area we will examine will be that of architecture. Since product evaluations are based upon an examination of both the hardware and software, the primary issue is how the underlying hardware structure supports the advertised software features. It is noted that although the Criteria requirement is for the TCB to be conceptually simple, it is recognized that it may be complex in practice. The challenge to the evaluators then, is to be able to determine how an arbitrary, and perhaps unfamiliar, architecture satisfy the Criteria. The next speaker is Mr. Terry Cureton of control Data Corporation, who will discuss NOS as an example of "untraditional architecture," driven by perceived performance requirements.

Terry Cureton

One of the advantages of speaking near the end of a conference is that you have the opportunity to comment on what has gone before. So before my prepared talk I would like to make a comment on an underlying assumption I've been hearing at this conference.

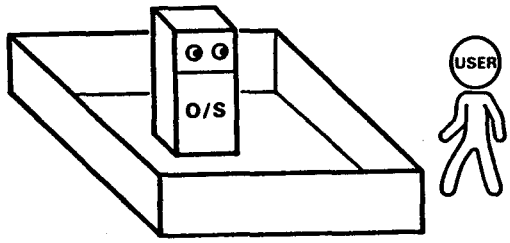
I think we have been focusing too narrowly on "data security" rather than on the broader subject of "computer security." This all hinges on the basic definition of "security" which is: the safeguarding of resources. Those resources include both data and computing resources. Thus we must deal with the denial of service problem to ensure the availability of both resources or we may find ourselves in a state of perfect data security where it is simply unavailable.

This is the fundamental problem addressed in the (B1) Security Testing section of the DoD Criteria which includes the objective: "...to assure that no subject (without authorization to do so) is able to cause the TCB to enter a state such that it is unable to respond to communications initiated by other users."

For this panel session, I would like to focus on one aspect of the DoD requirements, namely the effect of machine architecture on the trusted computing base (TCB) design.

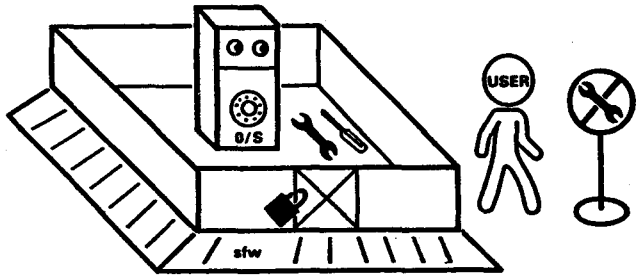
The term "non-traditional architecture" is simply a way of describing a machine architecture which is not a carbon copy of the more popular or traditional architectures.

ISOLATION/SEPARATION



The most fundamental aspect of system architecture relating to a TCB is the objective of isolating and separating the user from the operating system. Traditionally this has been provided by defining a security perimeter around the system and developing mechanisms to keep the user outside the system or TCB.

PROTECTION MECHANISMS



A number of hardware and software protection mechanisms have been used for this purpose, which I have tried to depict pictorially.

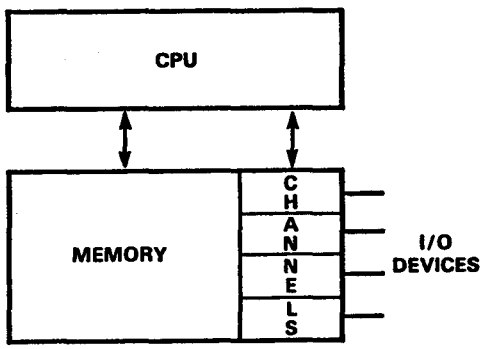
The tools lying within the perimeter and the sign near the user illustrate the idea of privileged instructions. In a two-state machine these are the CPU hardware capabilities which are taken away from the user.

The lock on the gate represents lock and key mechanisms by which some users are permitted into the system when in possession of the proper key.

The safe-like knob on the system represents the use of passwords and other combinatorial hurdles which will eventually allow any user to obtain access to the system.

Finally, there is the inevitable small mountain of software necessary to support these various mechanisms when they are not implemented or supported by hardware features.

TRADITIONAL ARCHITECTURE

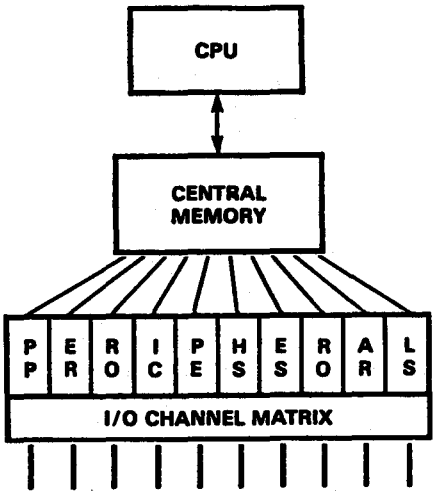


The key to the problem with traditional architectures lies in the arrows shown here connecting the CPU to the I/O hardware which permits both user and system CPU processes access to the real I/O environment. No matter how "virtual" the user environment is supposed to be, if the users have access to real I/O mechanisms they will eventually be able to penetrate the system.

A hardware architectural solution to this problem is found in the non-traditional architecture of the Control Data CYBER 17Ø series machines.

The row of boxes labeled collectively as peripheral processors, or PPUs, are small stored program computers

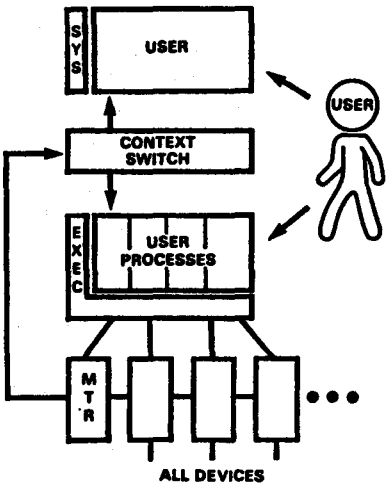
CONTROL DATA CYBER 17Ø ARCHITECTURE



which operate independently and concurrently with each other and the CPU. Only PPU processes can perform I/O and they execute only TCB software modules.

The PPUs provide the essential isolation of CPU processes from the real I/O environment which give the CPU users a virtual I/O environment and provides the TCB with total control of I/O.

CDC CYBER 17Ø USER/SYSTEM INTERFACE



Carrying this through into the user/system interfaces, we see where the user fits into the system.

The top box represents CPU utilization over time and illustrates that the user has the larger share of CPU utilization. Actually, system CPU utilization is typically less than 1Ø%, but the label wouldn't fit if accurately depicted.

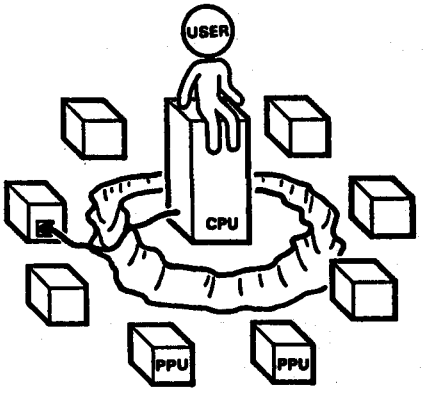
The third box down represents central memory space utilization and shows both the residence of the small system CPU executive and tables used to support PPU processes. Naturally, user processes in central memory are separated and protected from each other by a simple memory protection scheme in the hardware.

The second box between the CPU and central memory represents the hardware context switch mechanism we call an exchange jump. This is the mechanism used to switch the CPU between monitor and user mode and between processes in memory. It is totally transparent to user processes and very fast, roughly the same as a single floating point divide instruction.

At the bottom, the PPUs are in their usual place supporting I/O and other system functions. The one function singled out here is that of the PPU monitor (MTR) module which acts as the hub of the PPU subsystems and as a partner to the CPU executive. As a permanent and dedicated process, MTR keeps track of time and resource utilization.

The arrow connecting the PPUs to the context switch represents the hardware capability for a PPU to initiate a CPU context switch. Thus, a PPU has a hardware veto on any CPU process, including a system process, although in practice only user mode processes are context switched by PPUs. This is the mechanism by which a fine qranularity of resource controls are enforced by the system. Regardless of what the user is doing in the CPU, a system module in a PPU can always "pull the plug" on the user process.

CDC CYBER 17Ø ISOLATION/SEPARATION



This situation could be conceptualized by putting the user in the center, on top of the CPU and surrounded by the operating system. Here a user is free to utilize the power of CPU hardware, but is limited strictly to the manipulation of information in their assigned memory. This is a highly abstracted execution environment with very limited interfaces to the operating system.

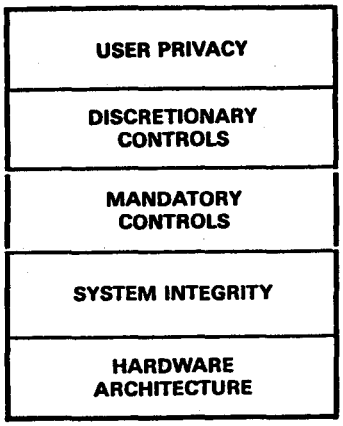
Although there are a few system CPU processes which are separated from the user by a conventional two-state CPU mechanism, the bulk of the system consists of PPU processes which are separated from CPU processes by the hardware separation inherent in the machine architecture. This separation is symbolized by the ditch around the CPU.

The multiplicity of PPUs (up to twenty) is symbolized by the ring of PPUs surrounding the isolated CPU. This represents their independence and concurrency of operation.

Finally, a symbolic CPU power cord is plugged into a PPU to represent the responsibility of PPUs to enforce resource controls and control CPU utilization.

Thus, the CPU user is thoroughly locked into a highly abstracted environment and completely dependent on independent external processors to provide essential services including the use of the CPU itself. Conceptually, this confinement of the user within the system is the inverse of trying to exclude the user from the system. This is the same principle used in our legal system for the confinement of people to prevent them from commiting further crimes. It is much more effective in computer systems because we can confine all of the system's users.

CONTROL DATA CYBER 17Ø NETWORK OPERATING SYSTEM (NOS)



It is on this hardware architecture base that Control Data has built the Network Operating System. The system integrity built on this base reflects the inherent strength of the architecture and more than 2Ø years evolution of the system software design.

At the top, we see user privacy as the primary objective of our system. After all, the original objective of multi-user systems was to provide the users with the kind of computing privacy now available via personal computers, but with the ability for controlled sharing of resources and data. The objective then is that a user should not be aware of the activities, or even the existence, of other users on the system unless they choose to share some information.

To meet this objective, the system had to have the capability for isolation and separation of users which we have already discussed, and basic mechanisms to enforce discretionary access controls based on information ownership.

Over the years, a number of our customers have noted that these access control mechanisms could be adapted to enforce mandatory access controls to meet DoD-type security requirements. Encouraged by their success, and by the DoD Computer Security Initiative, Control Data has undertaken to implement these capabilities into our standard NOS system.

What this entails is to implement DoD-type labeling throughout the system and to figuratively jack up the discretionary access controls and insert a layer of mandatory access controls into the system design. Although this is a relatively straightforward process not involving major structural changes to the system, it was non-trivial in the number of interfaces impacted. That's why it has taken six years to evolve MLS into our system design, spanning several system releases, and represents a major investment on the part of Control Data. Thus I have to agree with an earlier speaker, George Jelen, that such systems are slow in coming.

CONTROL DATA CYBER 17Ø NOS VERSION 2.2 RELEASE

- **STANDARD SYSTEM**
- **RELEASED 14 OCT. 83**
- **MULTILEVEL SECURITY (MLS) FEATURES**

That brings us to the Version 2.2 release of our standard NOS system. This system was released on 14 October 1983, within days of the distribution of the Final Criteria by the DoD Center.

Although there are many new features provided by this release, perhaps the most significant are those we call Multi-level Security (MLS) features.

CDC NOS V2.2 MLS HIGHLIGHTS

- **MARKING**
  - **LEVELS (8)**
  - **CATEGORIES (32)**
- **USER CLEARANCES**
- **MANDATORY CONTROLS**
- **SECURITY ADMINISTRATOR**
- **OPERATOR CONTROLS**
- **SYSTEM MODES**

For the sales portion of my presentation, I would like to describe some of the highlights and the bottom line benefits of our NOS Multi-level Security features.

As mentioned before, the major design impact of the MLS features is the marking of information with DoD-type labels. For this we define the usual eight hierarchical security levels and 32 non-hierarchical categories.

Users may be authorized access to some combination of these levels and categories via security clearance data stored in their validation records.

The third leg of the three-legged stool supporting the MLS feature is the mandatory security controls based on information marking and user clearances. The NOS MLS mandatory controls include both access and flow controls.

Although security levels associated with data are handled in a hierarchical manner by the flow controls, user clearances by level are actually treated as a bit vector and handled as discrete authorizations. Thus to create or access information at a given security level the user must be explicitly validated for that security level. This allows the possibility for an installation to define multiple hierarchies

of security levels, with users validated to different subsets, and yet have mandatory flow control. The classic example would be for interleaved NATO and US security levels, which would allow users to be cleared for US-only, NATO-only, or both, and with varying degrees of clearance. The design principle, which you will see in other areas of the system as well, is to design in as much flexibility as possible.

Concepts new to our NOS system are that of the system security administrator and the greatly tightened control of system in-house, our operators have found they have even less to do than before, yet are able to keep the system running smoothly. This tends to call into question why they needed all those controls anyway. The busy person now is the validated security administrator who must be called in to deal with all security-relevant controls on the system.

Since NOS is designed to be used by our entire customer base there will undoubtedly be installations where the MLS features and DoD-type mandatory security controls are less than desirable. For those sites, the MLS features can be turned off at system deadstart. In this mode, which we call unsecured, the mandatory access and flow controls are turned off, leaving the traditional discretionary controls which are compatible with earlier NOS versions. Information labeling features remain and NOS will still recognize and require operator approval to mount devices containing classified data. This covers the case of an unsecured mode deadstart with classified data on peripheral devices.

CDC NOS V2.2 MLS BOTTOM LINE

- **MULTILEVEL CONTROLS**
- **COMPATIBILITY**
  - **HARDWARE**
  - **SYSTEM AND PRODUCTS**
  - **USER PROGRAMS**
- **STANDARD**
  - **SYSTEM**
  - **SUPPORT**
- **FULL PERFORMANCE**

For some installations, the Multi-level Security controls are the primary benefit from NOS 2.2 and they would accept other tradeoffs to get them.

For our existing customer base, compatibility may be an important, or even critical requirement. NOS 2.2 MLS is both hardware and software compatible. It is downward compatible to earlier hardware, from the current CYBER 17Ø/8ØØ series through the CYBER 7Ø series, and even to the 6ØØØ series machines. The system software is upward compatible from previous releases to permit relatively easy system upgrades.

Since the user interface to the system is the least impacted by MLS features, the compatibility of software products is very high. User programs are typically compatible at the object code level so that most programs need not be recompiled. The bottom line is that most users will not be aware of the transition to NOS 2.2, as many of our internal users weren't.

However, the key word for MLS is standard. Especially for those customers who have built their own versions of MLS and had to do their own system upgrades, your suffering is over. The MLS features are a permanent part of our standard system and will continue to be supported by Control Data.

But the bottom line of interest to all customers is the cost for Multi-level Security. In this respect Control Data is different. Since we were unable to provide a performance degradation with the MLS, we could not justify charging more for these features. That is Control Data's answer to this morning's keynote speaker, Steve Walker, on the cost of computer security.

CONTROL DATA NOS DOD CSC
EVALUATION No. 1

- **INFORMAL EVALUATION**
  - **SINCE MAY 1981**
- **BASELINE NOS 2.0**
  - **COMPLETED**
- **PRELIMINARY NOS 2.2**
  - **UNDERWAY**

As I said earlier, MLS has been in development for six years, which is about the same length of time these DoD Initiative seminars have been going on. Since May 1981, we have been involved with the DoD Center with informal evaluation of our system design. So far, we have completed a baseline evaluation of the NOS 2.Ø release, which did not have the MLS features. Bill Neugent's description of this process as a "blind date" is most apropos. Looking back, that's exactly how it felt.

That first part is over now and a preliminary evaluation of version 2.2 with MLS is currently underway. The results of that evaluation will determine our future evaluation plans.

But for the benefit of those who have yet to sign up for that first blind date, I would like to mention some of the difficulties encountered and corresponding benefits of this process.

CONTROL DATA NOS DoD CSC
EVALUATION No. 2

- **DIFFICULTIES**
  - **ARCHITECTURE**
  - **INTERPRETATION**
  - **EVOLUTION**
- **BENEFITS**
  - **GENERALIZATION**
  - **DEFINITION**
  - **FEEDBACK**

First, I would point out that few things which are worthwhile are all that easy. The "difficulties" we have encountered in the evaluation process fit the classic

definition of a problem: "a perceived difference between expectations and reality." This is very close to Bill Neugent's description of the blind date.

The first perceived discrepancy is in the area of system architecture. It is clear that in the early versions of the Criteria, DoD expectations were for a system architecture based on kernel technology. From what I have presented here, I hope that it is evident how inappropriate this is for our non-traditional architecture. Yet, the fundamental principle of defining a TCB via modular system design is sound. Modularity is the basic means of managing complexity, such as is found in large scale systems, and a kernel is simply the end case of a single module. Thus the corresponding benefit from this difficulty is a noticeable generalization of the Criteria to permit application to both kernelized systems and modular systems regardless of machine architecture.

The second difficulty, to be expected in any process of defining and applying standards, is interpretation. We admit that our interpretations of the Criteria and implementation of features based on those interpretations did not always meet the Center's expectations. But via the evaluation process, we have come to a better definition of the intent behind the words, and the words have become more specific in the final Criteria.

The third difficulty, also inevitable, is that of evolution. Not only has our system design evolved to meet our interpretation of the Criteria requirements, but the Criteria has also evolved. This is compounded by the fact that our MLS design was essentially completed before we received the final draft Criteria, and there were significant changes between that version and the final Criteria. This is the classic moving target problem when you try to get a product to market before the market is precisely defined. Nonetheless, the evaluation process has provided valuable feedback on the merits of our design as well as the degree of compliance to the Criteria.

CONTROL DATA NOS DoD CSC
EVALUATION No. 3

- **PLANS**
  - **NOS 2.2 USAGE**
  - **DOD EVALUATION**
  - **NOS EVOLUTION**
- **GOALS**
  - **DOD CERTIFICATION**
  - **(B3) RATING**

So where is Control Data going from here? Our plans are straightforward. The NOS 2.2 system with Multi-level Security features is out on the street and will be used by most of our customers who will undoubtedly provide a lot of feedback.

We will persist with the evaluation of our systems by the DoD Center. At the same time we will continue the evolution of the NOS system design to meet both the needs of our customers and the requirements of the DoD Criteria.

In this regard, the goals to which Control Data is committed are to achieve security certification by the DoD Center and to obtain a class (B3) evaluation rating.

Mr. Paul Cudney is from Systems Development Corp., in Santa Monica, CA and will speak on the Kernelized Virtual Machine (KVM). KVM was a government-funded effort to develop a kernel-controlled IBM 370 VM system, along the lines of the virtual machine monitor concepts.

Paul Cudney

1. KVM/370 SYSTEM OVERVIEW

Some of you may not be familiar with the objectives of the KVM program, so I'll spend a couple of minutes mentioning some of its design requirements. The major program requirement was to support DoD security policy; the result in KVM was four security levels and 62 compartments. KVM also has discretionary access control lists and multiple passwords. Multiple passwords are used during LOGON to minimize spoofing attacks and compensate for the lack of a secure attention key; we require the KVM system to authenticate itself to the system. Access to objects follows the simple security condition and the "star-property" with one minor exception - KVM does not allow blind write-up. This intentional departure from the DoD security model is allowed by our formal specification language; it requires the user to specify the security model instead of assuming it in the specification system.

KVM implements a reference monitor, the security kernel shown below. It fits between the machine and everything else. There are some penalties in terms of performance.

REAL COMPUTER
REFERENCE MONITOR (SECURITY KERNEL)

NON-KERNEL (U)	NON-KERNEL (TS)
CONTROL PROGRAM	CONTROL PROGRAM
USER OPERATING SYSTEMS	USER OPERATING SYSTEMS

KVM is based on an old version of VM/370 - release 3. VM has evolved a great deal since that time, as evidenced by the recent release of the specifications for VM/SP release 3. There was a conscious decision made a long time ago to stabilize on one base. We are looking at the repercussions of that decision now.

The Non-Kernel Control Program (NKCP) shown in the figure represents much of the original VM Control Program. The original VM Control Program has been modified to work with the security kernel, instead of with the real hardware, whenever privileged hardware access is needed. KVM creates a separate copy of the Non-Kernel Control Program for each active security level. Each NKCP supports multiple user virtual machines, all running at the same security level. It is at this point where the original VM/370 architecture is probably more evident, and this is where conventional operating systems run. KVM supports non-virtual operating systems at this time.

Binary software copied from VM to KVM will work unchanged. It works since KVM preserves the VM interface, consisting of the System/370 hardware architecture and the CMS operating system. Software moved to KVM runs as it would under VM, with the added

constraint that it runs at a particular security level in its own security level address space. Software running at one security level is prohibited by the KVM kernel from communicating with software running at other security levels in any manner counter to restrictions of the security condition and star-property. When data is stored on disk, each virtual machine views only that data the security policy permits it to see, even though all data resides on the same physical device. All system data is thus under complete control of the KVM security kernel.

KVM does not depend on any hardware modifications to the System/37Ø machine. We suspect that had we an option to modify the hardware, we would have been able to improve performance somewhat. This ends the brief overview of KVM's architecture.

2. PROGRAM STATUS

Right now SDC is involved in a review effort to evaluate the performance of the system. We are attempting to provide a list of lessons learned from our experiences; standing outside of IBM, trying to jack up VM and slipping what was originally going to be a very small and simple security kernel in underneath. It turned out to be neither small nor simple. The system is quite large, its performance leaves a great deal to be desired in our own minds, and we are still studying it at this time. We hope to have the results of our study sometime about April of next year, and expect to be talking about KVM's performance at the IEEE Symposium on Security and Privacy in Oakland.

As a research prototype, which is really what KVM was, we met our goals. We showed it was possible to take an existing operating system, raise it up, put a kernel underneath, and mediate all access to security objects. KVM demonstrated that kernel technology can work, but does not yet work efficiently. We found hardware and software features that had to be disabled. For example, some ISAM software used self-modifying Channel Command Words, presenting a security problem that could be challenging.

The lessons we learned will be available; we hope they will be a matter of public record. SDC has found formally specified operating systems to be definitely non-trivial. We are applying these lessons to our own work, and are seeing the Ina Jo (Ina Jo is a trade mark of SDC) and ITP tools gain widespread use.

3. CONCLUSION

One of the things we believe the government was seeking when KVM was originally funded was for manufacturers to take the main responsibility for building security into their product line. It is pleasing to see some manufacturers doing this now. I would like to believe that our experience with KVM will provide some insight to others so that their job will perhaps be a bit easier in the future.

Our final speaker is Mr. Lester Fraim of Honeywell, who will present a status report on the SCOMP system. The SCOMP system is a modified Level 6 minicomputer which incorporates formal design and verification techniques, and is currently under evaluation as a candidate A1 system.

Les Fraim

One of the reasons I wasn't going to talk about SCOMP today was that when Chuck Bonneau and I came to the first session on Tuesday, we discussed that there had been five seminars and either he or I had discussed SCOMP at every one. So we figured that maybe everybody was tired of hearing about it, however, we also heard that 6Ø percent of the people are new and SCOMP was mentioned several times during discussions of verification and evaluation. So, I thought it was an opportune time to discuss the SCOMP program.

When you develop a product you must not only solve the technical problems, but you have to provide the marketeers something to talk about. We came up with a logo which you see behind me, consisting of the keyhole which is significant in the world of computer security. There's another logo that has some significance and this is the question mark. How do you build a trusted product? It's not easy! I agree with many of the things that have been said here today, and I also disagree with a lot of things that were said here on Tuesday morning. I think there are problems out there that have to be solved. How do we do it? I'm not going to spend a lot of time on the details of the reference monitor. As Paul didn't, you can read the pictures behind me. If you don't understand them, you can read the IEEE Computer magazine of July 1983. It has a great deal of description of the SCOMP, all the technologies that Carl Landwehr mentioned, and a nice article by Roger Schell about security kernel technology. The point I'd like to make is that when you talk about building secure systems, and SCOMP was an effort to see if we could do this with hardware and software, as was pointed out by some of the other speakers, the key is the architecture, without hardware support for security features in software, don't even try it. It's hard enough to do when you have hardware support. There are things that you must do in hardware to support mechnanisms to allow you to run fast enough that people will even want to use the machine. What's fast enough? I wish I knew. Depending upon what you use the SCOMP for it will provide various levels of performance. What do we say about the performance of SCOMP? We have minimized the performance degradation in a reference monitor security kernel based system. What's minimize? I don't know; that's a good word. That's like secure. What's secure? Depending on what you do, you can attain various levels of performance. Performance degradation is caused by the mediation requirement, and we try to minimize it through the hardware/software combination of our security kernel and our basic architecture that enforces the mediation mechanism.

The key point about this slide is the word product. This year, in August, we finally got Honeywell to approve SCOMP as a product. We will be in this year's GSA schedule as a class II software product and as a hardware product. It takes many people to get all the signatures necessary to allow you to call something a product in Honeywell. That's a major step for us because we now have taken a research and develpment effort and have transferred that technology into what Honeywell feels is necessary to market a product. The product is both the hardware mechanism and the software mechanism. The hardware features can be found in various parts of the

literature. The hardware is the base without which we would not be here today. We probably would not have been here four years ago or anytime before that. We are still fighting with some problems in the hardware. I wish I could say that we had them all fixed right now, but we are very close to having the ones we know about fixed and that will give us the ability to complete the evaluation process with the Center.

What is the hardware? Well, you may hear about the Security Protection Module (SPM). That is the unique piece of hardware we add to a DPS6 or a Level 6 that allows us to build a security kernel based operating system. All of the details are quite lengthy. There are volumes of documentation on the SPM. It provides a mediation mechanism which is descriptor based for both I/O and memory management capabilities. The virtual I/O is rather unique because it allows us to have I/O drivers outside of the kernel. It makes our kernel smaller and easier to convince the Center that it is really a kernel. It also allows us to be more efficient. The I/O mechanism in SCOMP is much more efficient than having to call a security kernel or an operating system everytime you want to do an I/O. One of the major things we've been able to minimize is the performance penalty for I/O. This reduction is because of the way we built the hardware architecture.

Now, on top of the hardware we built an operating system, which we again have labeled. I agree with Stan that if we don't have three or four letters for everything, you can't sell it, so we call the OS STOP. Again, trying to build in the knowledge of security. STOP stands for the SCOMP Trusted Operating Program. It provides a general 16-bit mini-computer operating system with all the bells and whistles of multi-level security. And all the bells and whistles are the difficult part, when it must run effeciently, provide capabilities and allow you to have administrators, operators, and users in your system. Now, we go around talking about the operating system, of course, the big difference between this operating system and the one you buy for commercial hardware is that it enforces mandatory security policy. Both security and integrity are enforced and we've all heard about the eight security levels and 32 categories. We also have the same number of levels and categories for integrity which makes things much more complicated. I'd like to reiterate at this point what has been said a couple of times and I can't emphasize it enough, it's not just to sell SCOMPs, people don't know how to write applications on trusted systems. Most people don't know how to do anything on trusted systems because of the lack of understanding of how the model is enforced by the hardware and the software. We have even fooled ourselves a few times thinking something would be easy and it turned out rather difficult because all of a sudden a simple program ends up not being able to talk to the right people and it becomes trusted. One of the areas which must receive attention within the next couple of years is application experience on a trusted system. How do we use them, and how do we use them effectively, so that we can meet the various requirements of government and private industry.

Four simple rules, they look simple up there, they are not simple when you are dealing with them. One of the things we ran into the first time we turned on all the categories - system high becomes something like TOP SECRET with 32 categories and when you print that at the top and the bottom of every page of printed output, you don't get much printed output. Things like that make it not nice, but it still must be done. The way we get around that problem is by making sure the categories used have very short names like 1, 2, 3, 4 so that they can fit on the page without wiping out everything you want to print.

The integrity rules are enforced and used extensively in the protection of the various features of the operating system and control of administrators, operators, and users.

I've been asked by many people in the audience, where are we in our evaluation with the Center? We are in the middle of it because of the moving requirement, we've only received the new criteria recently, and we are trying to identify where we are not in compliance. Discretionary access appears to be a problem area which we are attempting to solve. This is an interesting case of being developed under government guidance. The government that guided us is obviously is not the same Department of Defense we've been talking with recently. It is hard to go back and re-do some things that you have done six years ago. However, there are ways in which we think we can make some minor adjustments to meet the letter and intent of the Criteria so that we can achieve the A1 level from the DoD Center. The reason there is not much being said about where we are in the evaluation is that the final criteria is so new that we're really not sure where we are. It's different from going and looking at a C1 or a C2 system. You've got to remember that the Criteria is additive and all those pages and pages of requirements go into the back to form the A1 requirements.

Trusted software is almost totally specified in GYPSY right now and we are in the process of trying to run it through the verification tool. I think there's a major challenge there to see if we can do that. Learning how to write GYPSYis not easy. I'm one of those, who's heroic, I've been beaten to death by four very talented individuals who I made write GYPSY because it is not an easy thing to do. Somebody said the other day it doesn't take a PHD to write GYPSY. No, I think it probably takes two or three and even then you cannot get a concensus on what the the GYPSY should look like. We've gone through, with the Center, four or five iterations about what is really entailed in a GYPSY specification for trusted software that resides on top of a verified security kernel. No one has really thought about that before, but it does propose some new problems that we have to address. The simplest part of the operating system is the part that we invented ourselves. Again, it's got an acronym SKIP. It's the SCOMP Kernel Interface Package. It is really outside the TCB. It is an untrusted applications interface that allows you to write programs and to do things you need to do for an application. It was designed with government input and so we found that we constantly want to add a few things here and there. It doesn't do everything we thought we wanted it to do in the beginning but it provides a sufficient environment.

Applications - yes, there are applications you can run on SCOMP. It is not a rich full set of applications because we didn't listen to everybody. We didn't build our system to be compatible with anything else. It's compatible with itself. We do have a compiler and assembler and TCP/IP

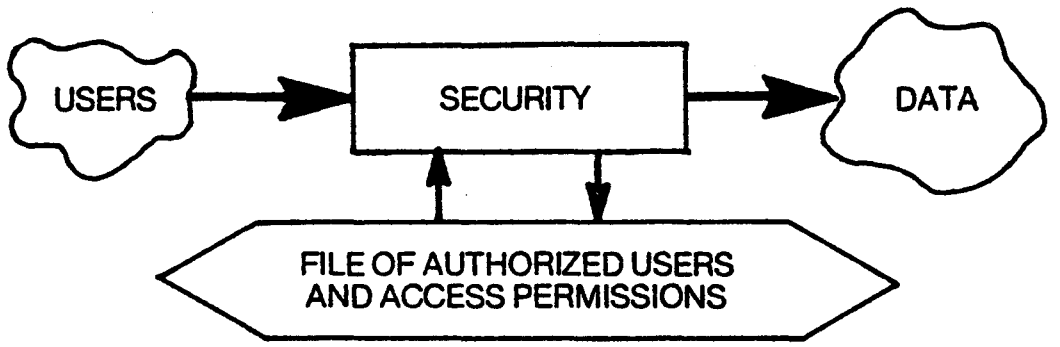
for the DoD community. We recently implemented an interface that allows SCOMP to act as a front-end to our large host. This is being used in an application development. So, there are capabilities that are useful. We have demonstrated a SCOMP at seven trade shows including the AFCEA and FCC in Washington and various at Honeywell trade shows. It will be at AFCEA show this summer in Washington which provides a good opportunity for people to see it, touch it, do anything they want to with it. The Center has a SCOMP. We have one in McLean so anybody who would like to get their hands on one can let me know.

What makes it trusted? I put this one in just to talk a minute about verification. I was glad that the verification session yesterday discussed many of the problems with verification. An A1 system is listed as within the state-of-the-art. I think it is within the state-of-the-art, however, I don't think it is something that everyone who works for a manufacturer can convince their bosses that they ought to do. It does provide additional assurance. It provides benefits when done properly, however, there are still a lot of questions about technology and I think the Center's role of trying to get some tools on systems that they can control and that vendors can use is the correct direction to meet that requirement. We still have fights in our own company of getting people to write in high level lanaguages. I think that Dan Edwards said to me at one time he sure wouldn't want to evaluate an A1 system in assembly language. I don't think I would either. SCOMP is in high level language and lends to the ability it has.

Now for my marketing pitch. What is the state-of-the-art? I don't know. We might as well say that SCOMP is. It is close enough. Everybody works to try to build something new and better. Chuck Bonneau has been on this project six or seven years now. I've been on it four. I set a goal that I'd make it to ten years with Honeywell working on SCOMP. I passed that last Saturday. I'm still here. We will continue to push the state-of-the-art with this kind of product. Thank you.

PRINCIPLES OF SECURE SYSTEM

- COMPLETE MEDIATION
- ISOLATION
- CERTIFICATION/SIMPLICITY



Slide 1

THE SCOMP PRODUCT

- BASED ON COMMERCIAL PRODUCT
  - LEVEL 6
  - DPS 6 (16 BIT)
- DEVICE COMPATIBILITY
- ASSEMBLY LANGUAGE COMPATIBILITY
- IMPLEMENTED IN PASCAL, C
- MULTICS ON A 16 BIT MACHINE

Slide 2

SCOMP HARDWARE

- DPS/6 - LEVEL 6 - 16 BIT
- MOST PERIPHERALS SUPPORTED
- ADDITIONAL HARDWARE
  - ENHANCED CPU
  - SECURITY PROTECTION MODULE (SPM)
- ARPANET INTERFACE TO DPS/6

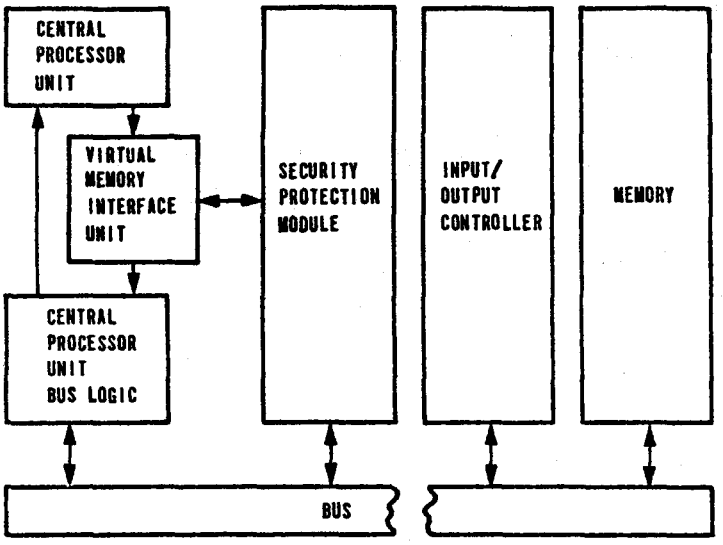
Slide 3

SCOMP HARDWARE OVERVIEW

- SCOMP HARDWARE CONSISTS OF A STANDARD MINICOMPUTER (HONEYWELL LEVEL 6) ENHANCED BY A SECURITY PROTECTION MODULE (SPM)
- FEATURES
  - MULTICS-LIKE RING STRUCTURE
  - RING CROSSING SUPPORT INSTRUCTIONS MEMORY MANAGEMENT
  - MILLION WORD ADDRESS SPACE
  - PAGE FAULT RECOVERY SUPPORT
  - FAST PROCESS SWITCHING

Slide 4

SPM + LEVEL 6 MINICOMPUTER = SCOMP



Slide 5

SCOMP TRUSTED OPERATING PROGRAM (STOP)

- SECURITY KERNEL-BASED SYSTEM
- TRUSTED USER INTERFACE
- APPLICATIONS INTERFACE
- SECURITY ADMINISTRATION FUNCTIONS
- OPERATOR COMMANDS

Slide 6

SCOMP

SECURITY POLICY

- SUPPORTS MANDATORY POLICY
  - SECURITY
  - INTEGRITY
- SUPPORTS NEED-TO-KNOW (DISCRETIONARY POLICY)
- PROVIDES TRUSTED TERMINAL INTERFACE
- SUPPORTS A HIERARCHICAL MULTI-LEVEL FILE SYSTEM
- SUPPORTS PROCESS COMMUNICATION AND SYNCHRONIZATION

Slide 7

SCOMP

MANDATORY POLICY

- SIMPLE SECURITY - NO READ UP
- * -PROPERTY SECURITY - NO WRITE DOWN
- SIMPLE INTERGRITY - NO READ DOWN
- * -PROPERTY INTERGRITY - NO WRITE UP

Slide 8

SCOMP

DISCRETIONARY POLICY

- READ, WRITE, EXECUTE FOR OWNER, GROUP, OTHER
- RING BRACKETS FOR OWNER, GROUP, OTHER
- SUBTYPES

Slide 9

SCOMP

TRUSTED INTERFACE

- TRUSTED PATH TO SECURITY SENSITIVE SOFTWARE
- USER-TERMINAL INTERFACE
- ADMINISTRATOR FUNCTIONS
- OPERATOR COMMANDS
- MECHANISM FOR CONTROLLING APPLICATIONS ENVIRONMENT

Slide 10

SCOMP KERNEL INTERFACE PACKAGE (SKIP)

- MULTI-LEVEL HIERARCHICAL FILE SYSTEM
- PROCESS CREATION AND SYNCHRONIZATION
- I/O SUBROUTINE PACKAGE

Slide 11

WHAT MAKES SCOMP TRUSTED?

- DESIGN VERIFICATION
- IMPLEMENTED IN HIGH LEVEL LANGUAGES
- ENFORCES MANDATORY SECURITY & INTEGRITY
- HARDWARE/SOFTWARE REFERENCE MONITOR

Slide 12

THE EVALUATION PROCESS AND PROBLEMS

Paul Woodie

DoD Computer Security Center

Abstract: *This paper describes a major goal of the DoD Computer Security Center, which is to encourage the easy availability of computer products with enhanced security features. The mechanisms by which this is to be accomplished are described. There are detailed explanations of the Developmental and Final Product Evaluation processes. The paper then takes a pragmatic view, from three perspectives, of how the process is actually working. Finally, an update is included, which describes the present status of the evaluation efforts underway.*

INTRODUCTION

For reasons of economy, efficiency, and a whole host of other reasons, there is a need, on the part of the Department of Defense, for computer systems that can operate in the "multi-level security" mode (users at multiple security levels simultaneously processing information at multiple classification levels). Indeed, there appears to be a similar need on the part of commercial organizations as well for some sort of multi-level secure processing. It is the policy of the DoD to encourage the easy availability of trusted computer systems. As one of the steps in implementing this policy, the DoD has formed the Computer Security Center (the Center). One of the goals of the Center, the one we will consider in this paper, is to encourage computer vendors to provide, on an off-the-shelf basis, computer systems with enhanced security features which can support the DoD security policy. One of the the mechanisms through which we plan to achieve this goal is that of computer product evaluation.

In the past, the DoD, in order to meet its requirements for secure computer systems, found itself specifying to the computer vendors specific security features that were required for the systems they wished to procure. Over time, the DoD found that it was specifying again and again, to the computer vendors, how to build secure operating systems. This was really not good for either the DoD or the computer vendors. It was not good for the DoD in that they were "re-inventing the wheel" each time they needed a new computer system. It was not good for the vendors, since they, not the DoD, knew more about how to build operating systems. To get the DoD out of the largely non-productive loop of designing operating systems, and to more clearly communicate to the vendors what kind of secure, multi-level computers systems were needed, the Trusted Computer System Evaluation Criteria were prepared.

The Trusted Computer System Evaluation Criteria (hereinafter called the Criteria)[1] has been described by Schell,[2] and included in that description is a brief explanation of the security provided in each class and division of the Criteria. This paper presents a review of how the Criteria have been used in the evaluation process that the Center is now undertaking. Included in this review will be a presentation of the industrial relations program that the Center is conducting, the two types of evaluation relationships with vendors that the Center maintains, three different perspectives on how the processes are working, and the status of the evaluation efforts to date.

I would like to stress here that the evaluations I refer to are strictly evaluations of off-the-shelf computer equipment and not of computer systems in specific applications. Hence, we are dealing with the computer system itself and the security properties built into that system, and not with any administrative or other procedural security features attendant to any specific system.

Vendor/Industrial Relations Program

As stated earlier, one of the overall objectives[3] of the Center is to encourage the easy availability on an off-the-shelf basis of computer systems with enhanced security properties. One of the the mechanisms that the Center has chosen to do this is that of evaluating computer products against the Criteria and publishing the results. These published results will be made available primarily to acquisitions people in DoD (although they will also have a wider availability through the National Technical Information Center). It is anticipated that this process will have at least three benefits. First, it will enable acquisitions people to specify more clearly the security-related computer products they are acquiring. Not only will they be able to better understand what is available in the marketplace, but also they will be able to more clearly define to the vendors what it is that they wish to procure. The second anticipated benefit is that the vendors will have, even before any specific acquisition process, a better idea of what kind of security features and assurances the DoD wants in the systems they use. The third anticipated benefit is a direct consequence of the first two; i.e., vendors will be encouraged to build enhanced-security products which can be made available to the DoD (and other customers) on an off-the-shelf basis. Note that the whole idea here is to elicit willing cooperation of the vendors in this process.

There are two basic types of evaluation relationships that the Center will maintain with a vendor: preliminary (or informal) and final (or formal). In most cases, the process is initiated by the vendor. When a vendor decides that he would like to build a computer product with significant security properties such that it would achieve a given rating on the Evaluated Products List (EPL), at some point in the early stages of the design process he may contact the Center to arrange for a "preliminary" evaluation. Both the Center and the vendor will then devote manpower resources to the developmental evaluation process. The basic purposes of the preliminary evaluation are to address security-related design issues at design time and, hopefully as a result, to arrive at a developmental assessment of where the proposed design would rate against the criteria. Keep in mind that the vendor at this point has made no committment to complete the product development or market the product.

The second major type of evaluation relationship that a vendor could have with the Center is a final (or formal) evaluation. In this case, the vendor has a product which he is either already marketing or plans to market in the near

future. In this case, the vendor can request that the Center initiate a formal product evaluation. In the next two sections we will examine each type of evaluation a little more closely.

Developmental Evaluation

As mentioned above, a preliminary evaluation is initiated by a vendor when he would like to obtain an initial estimate of how a potential product would be rated against the Criteria and what would need to be done to achieve a target rating. There is no committment on the part of the vendor to complete the product or bring the product to the marketplace. The decision to market a product must be made by the vendor, based on his own marketing research. The Center has no intentions of telling the vendor how to market or what to market. By providing to the vendor, during the early design stages, an initial assessment of how the potential product would rate against the Criteria, we at the Center expect to have an opportunity to influence the design of new, enchanced-security, computer products.

We will meet with the vendor to coordinate our joint activities, and out of that meeting will come a list of activities and rough timetable of activities that should lead to an understanding on the part of the vendor of the security issues involved in designing a product which would meet the target rating that he would like to achieve. Keep in mind, there is no commitment on the part of either the vendor or the Center to proceed with a formal evaluation. The vendor is free to withdraw from the evaluation at any time he desires. The entire effort is vendor-driven, not Center-driven or schedule-driven. At the end of the developmental evaluation, a wrap-up report will be produced. This wrap-up report, however, will result in neither a formal rating nor placement on the EPL. In addition, the report will not be distributed to the general public, nor even to the DoD at large, since, in most cases, it will contain proprietary data.

As a part of the preliminary evaluation process, a baseline working paper may be prepared by the evaluation team. This baseline working paper may be used as a vehicle for communication between the evaluation team members and the vendor's design team. This would, in the case of a product that is being upgraded, be focused primarily on the existing (prior to upgrade) product. The basis for this working paper would be any material that the vendor provides describing the product in as much detail as is necessary for the evaluation team to understand the product as it relates to the Criteria. A typical example of this would be the so-called internals courses offered by the vendors for their particular systems. In cases where there is no such formal course offering, some other information transfer arrangement would be arranged between the Center and the vendor.

Some time after the baseline working paper has been received by the vendor, evaluation team personnel will meet with the vendor's design personnel to discuss the paper, including how the computer system under evaluation meets or does not meet each specific requirement of the Criteria. This discussion, and any ensuing discussions, will point out to the vendor what he has to do to upgrade his system to achieve some higher Criteria rating that he may have set as a target rating that he would like to achieve.

The developmental evaluation will proceed, on a more or less informal basis, until either the vendor has obtained the information he is seeking, or until either the vendor or the Center decides that the developmental evaluation has served its useful purpose. It should be noted that the Center is fully prepared to protect any information that the vendor considers proprietary. To this end, the Center will execute with the vendor a Non-Disclosure Agreement.

Formal Product Evaluation

A vendor can enter into a formal evaluation relationship with the Center in one of several different ways. He may proceed into it as a natural result of the developmental evaluation along with his normal product development cycle. This would normally be the case where a product is either being upgraded or developed as a new product. He may also, for reasons best determined by himself, not desire (or indeed need) a developmental evaluation. In either case, though, the formal evaluation is initiated by the vendor request.

I should emphasize at this point that the vendor need not have a product on the market. As long as there are firm market plans, in the near future, and at least a field test release product which is available to the government, a formal product evaluation can occur.

The formal evaluation process is in some respects similar to the informal evaluation. For example, both evaluations are against the Criteria, both have an initial education phase, and both culminate in a final report. However, there are some very distinct differences. The largest differences are that the final evaluation report will be widely available to the public and the product will be placed on the EPL at whatever rating the product achieves. The vendor will, of course, have an opportunity to review the report to ensure technical accuracy and to remove any proprietary information. The Center, however, has firm plans to complete an evaluation, once begun, and to publish a final report and rating. There will be a firm schedule for the evaluation which will be based on constraints of both the Center and the vendor.

As part of the formal product evaluation, the Center will issue, as appropriate, Product Bulletins, which describe the product, and the candidate evaluation class against which the product is being evaluated. The Center to date has issued several of these Product Bulletins on computer products which either have been or are currently under formal product evaluation.

Depending on the product being evaluated and the class for which it is being evaluated, there will be a publicly available portion to the final evaluation report and there may be a limited distribution portion. The publicly available portion would contain a description of the security-related features of the product, and how the product was rated against the Criteria. The limited distribution portion of the report would contain information about any security weaknesses of the product that were discovered during the evaluation (primarily the testing phase) and any other information which the vendor considers proprietary. It is intended that this limited-distribution portion would only be available to the vendor, government users of the particular product, and potential government users (e.g., those within the government who have a demonstrated need-to-know).

Observations on the Evaluation Process

Now that we have examined the two types of evaluation processes, let us consider how the processes are actually working. Most of my remarks here will be focussed on the Formal Evaluation, however they also apply, to a lesser degree, to the developmental evaluation process. We will consider the evaluation process from three different points of view. That is we will consider:

- o the role of the Criteria in the evaluation process;
- o the role of the evaluators in the evaluation process; and
- o the role of the vendor in the evaluation process.

The Role of the Criteria in the Evaluation Process

As was pointed out in a previous paper, the Criteria reflect the results of at least ten years of research into the question of computer security. As a result, the basic concepts of computer security have evolved from some rather coarsely stated principles to a rather explicitly stated set of features and integrity requirements as embodied in the Trusted Computer System Evaluation Criteria. In the Criteria there are some requirements which, although they are rather explicitly stated, can be met by any number of different methods. A particular manufacturer, using a particular technology and a particular hardware/software implementation, may choose to implement some feature in one way whereas some other manufacturer would do the same thing in a different way. The decision on whether or not a particular implementation is acceptable should be based only on the requirements of the Criteria and not on the particular technology used, or other implementation details. The Criteria (dated 15 August 1983) places emphasis, to the maximum degree possible, on the security requirements and not on the actual method of implementation.

As we at the Center have gained experience applying the Criteria to real life computer products, we have run into a number of instances where it was necessary to apply some amount of judgement to the process of determining exactly what are the Criteria requirements in the context of a particular implementation. This is not at all unlike the situation that the United States Supreme Court is in. The Congress writes the laws and the Court has the responsibility of interpreting them on a case-by-case basis as the need arises. In this particular case the Criteria have been developed in an open forum with many participants. The particular part of the Computer Security Center that the author represents now has the responsibility of interpreting and applying it to specific implementations. Certainly, our goal has been, and will continue to be, to further sharpen our focus on requirements and reduce our focus on implementation details. There are a number of areas where the need for application of sound judgements has been, and will continue to be, required. Some specific examples are in the areas of auditing, labeling, the concept sometimes referred to as denial-of-service, and the whole area of specification and verification. For example, at what level of abstraction should the Formal Top Level Specifications be required for the A1 rating? As long as the specification (at whatever level) is stated in mathematically rigorous terms, should the level of abstraction be a primary point of concern?

These questions, plus others that have occurred (and will continue to do so), highlight the requirement for consistent application of sound judgements. Hence, as our experience in application of the Criteria continues to grow, we are developing a set of interpretations and applications guidelines that will enable us to apply the Criteria uniformly and consistently.

Another interesting reflection on the Criteria has to do with how the Criteria compare with the present computer marketplace. For example, most, if not all, existing computer products were developed prior to the Criteria. Most computer products have security features which fall at several different levels of the Criteria. There are few, if any, computer products that fulfill all of the Criteria requirements at any specific level and none of the requirements at the next higher level. For example, there are few, if any, computer products that completely meet all of the C2 requirements and none of the next higher level (B1). However, it is the intent of the Center to give a manufacturer credit in the evaluation for each and every feature that exists in the particular system being evaluated.

The Role of Evaluators in the Evaluation Process

It is important that the evaluators be as objective as possible in the evaluation process. To this end, the Center has endeavored to have heterogeneous instead of homogeneous evaluation teams. For instance, the present evaluation teams are composed of members from both operational organizations within DoD as well as technology-based organizations. The present product evaluation team examining the Honeywell Multics, for example, has members from the Computer Security Center, the Mitre Corporation, the Aerospace Corporation, the U.S. Air Force, and the University of Maryland faculty. The other formal product evaluations, depending on the degree of complexity and other factors, have similar compositions.

There are at least three types of computer science skills that can be used in the evaluation process. These are: experiential skills, theoretical skills, and formal skills. (This concept was first proposed by Epstein, Marsden, and Kramer of the Mitre Corporation.) Experiential skills refer to those skills that enable one to use the software facilities of the particular system being evaluated. The spectrum of these skills would include those who could only log in and out of the system and use only the simplest of applications packages to those who are system programmers, capable of using the system to its fullest and even modifying it as needed. Theoretical skills refer to those skills whereby one can understand the underlying modular structure and principles of the particular system under consideration. The spectrum here would include both those who know only the basic architecture and logical structure of the system and those who know not only the basic architecture of the system, but also know in detail how each of the modules interacts with each other module (both hardware and software). Formal skills refer to those skills that enable one to use the formal specification and verification tools that are required at the higher levels of the Criteria. The spectrum here would include those who could only

understand these tools and how they are used after a more skilled person interprets them. The other end of this spectrum would include those who could interpret these tools, apply them to specific cases, and even modify or create new tools as required. Although the specific set of skills required in any given evaluation will vary depending on the system being examined, the Center has made, and will continue to make, an effort to have as many of these skills as required represented on the evaluation team.

The Role of the Vendor in the Evaluation Process

The vendor plays a key role in the evaluation process in that it is the vendor that initiates the process. In fact, the largest single factor affecting the evaluation is the vendor. It is the intent of the Center that the Criteria be clear enough that the vendor would be able to arrive at roughly the same conclusion about the rating of his product as would an evaluation team. Furthermore, it is is the intent of the Center to supply as much information as is necessary to the vendor to enable him to properly interpret the Criteria. Hence, the role of the Center as a product evaluator is similar to that of a quality control organization. The Product Evaluation part of the Center would expect the vendor to come to the evaluation with all of the evidence in hand that will justify to the evaluators that the desired product rating is indeed justified. The vendor, given that he takes an objective look at his system, should be able to arrive at the same rating of the product as the evaluation team. Hence, the vendor should know in sufficient detail what will be required to justify the particular rating that he is seeking.

At some point after the vendor's initial request for a formal product evaluation, the Center will assemble a product evaluation team and arrange an initial meeting with the vendor. The purpose of this meeting is to make sure that both the vendor and the Center know exactly what items (e.g., documentation) and other supporting evidence will be required in order to justify a given rating for the computer product. At this point the role of the vendor is quite clear: he must not only produce such evidence, but also provide to the evaluation team members enough knowledge of his particular system so that the evaluators can properly understand the information and other justifying evidence that he has submitted. For a number of the evaluation classes it will be necessary to perform some amount of hands-on testing (security testing). Here again the vendor can play a significant role in either providing a machine or other test bed for these tests or in arranging or coordinating such a facility. Finally, the vendor will play a substantial role in the evaluation by providing a review of the final report(s) that the evaluation team will prepare. Again, this review is for the purpose of deleting any proprietary material or technical errors.

Present Product Evaluation Activity

At present, one Product Evaluation has been completed and there are additional Product Evaluations (both Formal and Developmental) in process. The Formal Evaluations include candidates for each of the evaluation divisions (A,B,and C). The table below shows the product, and the candidate evaluation class.

FORMAL PRODUCT EVALUATIONS

Product	Candidate Class
SCOMP (Honeywell)	A1
Multics "	B2
ACF2 (SKK)	C2
Top (CGA)	C2
Secret	
SEL (Gould)	C?

At present, there are also developmental evaluations being conducted with Digital Equipment Corp., NCR, and Control Data Corp. Assuming that all of these efforts, plus others which are now only in the initial inquiry phase, come to fruition, within the next three or four years there should be a number of enhanced-security computer products on the market.

Summary

In summary, one of the overall goals of the Center is to encourage easier, off-the-shelf availablilty of computer products with enhanced security features. The primary mechanism through which we plan to achieve this is product evaluation. There are two types of processes through which the Center plans to accomplish this: Formal Product Evaluation and Developmental Product Evaluation. The Center is now involved in both Formal and Developmental evaluations and anticipates having four computer products on the EPL by the end of FY83 and, potentially, a number of additional, security-enhanced products available in the next several years. In conclusion, in the past year we have come a long way toward our original goal of getting the vendors to supply, on an off-the-shelf basis, significant security products or enhancements, whose goal will be to support the DoD security policy.

REFERENCES

1. DoD Computer Security Center, "Trusted Computer System Evaluation Criteria," 15 August 1983.

2. Schell, R.R. "Evaluating Security Properties of Computer Systems," *Proceedings of the IEEE Computer Society, Technical Committee on Security and Privacy*, April, 1983.

3. DoD Directive, 5215.1, "Computer Security Evaluation Center," October, 1982.

COMPUTER SYSTEM SECURITY TESTING

Maj. Douglas B. Hardie, USAF
DoD Computer Security Center

Security testing is one of the techniques used by the DoD Computer Security Center in the evaluation of hardware and software systems. While testing can only prove that a particular system is not secure (presence of errors), the inability to find errors during testing does not prove the system correct. It only gives a measure of assurance that the system may be correct. The more effort invested in testing should lead to a higher assurance of correctness.

SECURITY TESTING HISTORY

Security testing has been used for at least 17 years to evaluate computer system security. During this time, the techniques of testing have been developed to improve its effectiveness and reduce dependence on abilities of a small group of individuals. The following history shows some of the major milestones in the security testing field. Note that this list is not complete since many testing efforts and results were classified or otherwise kept out of public purview.

AUTODIN I - Testing was accomplished around 1966 by NSA and was terminated when they felt the analysis was "good enough."

ADEPT-50 - Testing was accomplished primarily within SDC during 1968-70. One of the interesting techniques used was that of offering a bounty (cash) for penetrating the system.

IBM 360 - Testing by MacDonald Douglas around 1970 became an unending cycle of finding a problem, fixing it and then finding another problem. A key observation that a system cannot have security patched in was the obvious conclusion.

ANSERS - Testing by DIA during the early 1970's initiated the development of a subverter program. This is a program that periodically scrounges through the system attempting to penetrate the security mechanisms. Early detection of a mechanism failure limits the resulting damage.

The Princton Workshop in 1970 resulted in the definition of the Trojan Horse technique that has been successfully used to penetrate many systems.

DIAOLS - Testing by DIA around 1972 made extensive use of personnel from several government agencies. The original test plans called for the system developers to perform the penetration attempts. However, they ran into the developers fixing the system before running the official tests and resorted to independent testers to complete the testing.

MULTICS - Extensive testing by several groups around 1974 demonstrated effectiveness of planting a trap door in the source code for a multi-site system. Eventually you gain access to all sites.

VM/370 - Testing during 1974-75 by IBM and SDC used SDC's Flaw Hypothesis Technique.

Additional development work on penetration tools and generic flaws was done by Livermore Labs and USC-Information Sciences Institute.

TESTING REQUIREMENTS

The DoD Trusted Computer System Evaluation Criteria recognizes the usefulness of security testing and specifies requirements for testing of systems being evaluated. The requirements call for increasing testing for increased assurance.

Division C

- The evaluators will independently run test programs that were originally used by the developers for system checkout.

- The evaluators will test for obvious flaws or ways to bypass the security or audit mechanisms. At least five specific tests will be performed.

- Testing is expected to take one to three months.

Division B

- The evaluators will independently run test programs created by the developers to demonstrate security-relevant hardware and software operation.

- The evaluators will demonstrate that the system is found resistant to penetration. At least 15 specific tests to circumvent the security mechanisms will be performed.

- Testing is expected to take between two and four months.

Division A

- The evaluators will independently run test programs created by the developers to demonstrate security-relevant hardware and software operation.

- The evaluators will demonstrate that they are unable to penetrate the system. At lease 25 specific tests to circumvent the security mechanisms will be performed.

- Testing is expected to take between three and six months.

TEST TEAM

During a formal evaluation of a commercial system, the Computer Security Center will form a test team to conduct the test program. Since the Center does not have enough personnel to conduct all testing and evaluations internally,

we have contracted with MITRE and Aerospace Corp. primarily to provide assistance. In addition we often find users in DoD who are very familiar with a particular product and are willing to participate. While the Center manages the test effort, we expect to draw upon these resources to develop and conduct some of the testing.

The people chosen for the test team are carefully selected. The Criteria specifies the minimum backgrounds for team personnel: for Category C, members must be familiar with the "flaw hypothesis" methodology; for Category B, one member must have previously completed a security test; and for Category A, two members must have previously completed a security test. Most of the security testing experts that have become established through the many years of testing history are no longer available for the time-consuming efforts we require. Therefore, we don't expect to have the "gun-slingers" on every team.

TESTING FACILITY

The actual testing will have to take place on a machine that has the proper version of both the hardware and software to be evaluated. We frequently run into a problem here since the Center will never have enough computer hardware in-house to test all anticipated systems. Consequently we look elsewhere for a test facility. For those instances when we do have the hardware available, we prefer to conduct the testing in the Center. This is simplest for most members of the team and usually does not run into problems with other users of the system. In addition, we have arranged with other government agencies to use their facilities. This can cause problems since we then must work on a time available basis. It becomes difficult to hold to any test schedule in this environment. It would also be possible, but not preferable, to use a vendor's facility. The arrangements in this case would be complicated because of the nature of our team. Nevertheless, these things can usually be worked out.

Regardless of whose facility we use, we must have assurance that both the hardware and software are the correct version and configuration. In addition, testing requires the ability to use the front panel of the computer. While the actual penetration attempts are done as a normal user, sometimes it is helpful to have internal access via the front panel beforehand. And last of all, the facility must not be used for other work during the test times. If the team succeeds in penetrating a system, any other concurrent users would become upset; especially if the team succeeds (not necessarily deliberately) in crashing the system or corrupting another user's data. All of these factors complicate the location of a suitable facility.

THE TEST PROCESS

Once a facility is available, the team begins by becoming familiar with the system to be tested. Then they begin to establish a plan for both functional testing and penetration testing. Functional testing is performed first to verify that the system enforces the security policies that are claimed by the vendor. This is a demonstration, not an exhaustive test. Usually there is no formal test procedure developed, the team members take the features of the system and try them in various different ways. Test procedures or programs previously developed by the vendor will be used and the team may develop more if they feel so inclined. As a result of this testing and evaluation of system documentation, the team will develop hypotheses where the system security may be weak enough to penetrate.

The team then uses the hypotheses to develop specific tests to penetrate the system. These tests will generate additional hypotheses to be tried. This continues until the team decides to terminate the testing. Note that penetration testing is not exhaustive. Only a small number of hypotheses can actually be tested on the system. The team's challenge is to identify those hypotheses with the greatest probability of success. The testing purpose is not just to determine that there are ways to penetrate the system, but to identify as many ways as possible to penetrate the system. While the team will use the time frames in the Criteria as guidance for testing, they may extend testing some reasonable amount if they expect that significant areas remain that should be checked. The techniques used and their results provide a useful baseline for the systems developer for product improvement and future developments. This information will be made available to only the vendor at the completion of the testing, and to selected DoD users with special requirements to help them develop countermeasures.

PENETRATION TRAINING

Penetration testing requires a skilled team to effectively complete testing within the Criteria time frames. Currently the Center does not have a large resource of skilled penetrators. To overcome this deficiency, the Center has initiated development of a penetration test training program to build and maintain an effective penetration capability. This program is intended to be available for training new penetrators whenever they are needed. Personnel turnover and penetrator burnout will necessitate a continuing training program.

The training program is designed for small groups, although we believe that it could be useful for only one student. It begins with an academic phase where many of the previous studies and reports on penetration are examined. The theories (e.g. flaw hypothesis technique etc.) are explained and the student is led through the use of them. We envision that most of this phase is textbook type reading, but there needs to be an experienced training supervisor available to assist in this process.

At the end of this phase, we provide the student with a penetration handbook that is somewhat of a cookbook approach to planning the penetration of a new system. This book is intended to provide a handy reference to techniques as well as an attempt to insure that test planning does not overlook any major areas. We plan to have a "test" system available for the students to attack. This system will have its configuration locked so that it starts out the same for all students. It will also have several flaws that can be exploited by the students. The training supervisor will guide the students in the development of their test plan for this system. The purpose is to insure they will attack areas where they will succeed in penetrating the system and some areas where they will fail. The number of tests and the time spent is controlled by the training supervisor to provide appropriate experiences within the available time.

With this training program, the Center expects to be able to provide effective penetration testing throughout the coming years without reliance on the "gun-slingers." We have started the development of the training program and expect to have it in use by the end of this fiscal year. It will be available for the test team members, and can be made available for other selected DoD agencies. At this time, we don't see any significant benefits to wider dissemination.

The Center expects to use security testing as a significant tool in the evaluation of security for many years. Until program proving capabilities can be routinely applied to evaluating security at the source code level, security testing is necessary to achieve the desired levels of assurance in security. The Center is developing a security testing program that will insure the ability to effectively provide assurance via testing until that time.

EVALUATIONS OF APPLICATIONS SYSTEMS

Suzanne O'Connor

DoD Computer Security Center

Since June 1981, several of the Department of Defense Computer Security Center (DoD CSC)'s more important responsibilities have been:

1. Computer security evaluation criteria;
2. Evaluations;
3. Evaluated Products List;
4. Research and Development;
5. Focal point for computer security;
6. Technology transfer;
7. Point of contact between government and industry;
8. Consolidated Computer Security Program.

This paper will discuss the role of the DoD CSC Applications Systems Evaluations Office in meeting these responsibilities. The DoD CSC Applications Systems Evaluations Office, in concert with the DoD CSC's Standards and Products Office, the Research and Development Office and the Technical Support Office, contributes to the support of the Center's goals by providing technical guidance and assistance necessary to implement, assess and/or improve the security of developmental automated information systems and computers in operational systems which process sensitive information within NSA/CSS, DoD and its contractor facilities, and with other elements of the National Security Establishment. Furthermore, the Applications Systems Evaluations Office is required to make recommendations to the appropriate decision-making authorities regarding the operational use of these computer systems and attempt to improve the rigor and completeness of the design, implementation and evaluation of security in computer applications. Specifically, the Applications Systems Evaluations Office is tasked:

1. To provide technical support to system acquisition authorities in the selection, design, implementation and evaluation of hardware, software, and procedural security methods and techniques.
2. To conduct evaluations of selected computer applications, by request.
3. To assist in the certification and accreditation process for trusted computer systems.
4. To develop techniques, standards, and criteria for conducting evaluations and interpreting the results.
5. To establish and maintain technical liaison with other computer security evaluation organizations.

It is useful to remember that the Applications Systems Evaluations Office makes full use of the work and products from the Research and Development Office and the output of the Evaluated Products List in the solution of real-life computer security problems. We work with the user, from theoretical concept through the demise of a system. The Applications Systems Evaluations Office consists of two units, the Developmental Systems Evaluations Office and the Operational Systems Evaluations Office which evaluate proposed or developing systems and operational (or systems just prior to operational stage) systems respectively. Each unit is further subdivided into three subunits which evaluate NSA/CSS in-house computer systems, DoD computer systems, and National Security Establishment computer systems. The Applications Systems Evaluations Office is a service organization and only becomes involved in the evaluation of a system by request because we are a task-driven organization.

Once a task has been received and accepted, the evaluation process begins. If we have been tasked to evaluate a System Concept Plan or a System Acquisition Plan, an evaluator from the Developmental Systems Evaluations office studies the computer security-related portions of the plan. The evaluator will specifically look at the proposed mode of operation for the system, the type (classification and compartments) of information to be processed and stored in the system and any network interfaces. All pertinent DoD regulations and directives are considered as well as the Trusted Computer System Evaluation Criteria (hereafter, the Criteria). The evaluator must first decide what security policy must be adhered to, then the evaluator gives the system a preliminary candidate class (according to the Criteria). Determining a candidate class helps to set the evaluation framework for the proposed system. A candidate system for Class B3 and operating with users with access to two different classification levels must meet more stringent assurance requirements than a system selected for Class C2 where all the users have access to the same classification level. The evaluator then studies the system plan to assess how well the system designers have considered the security requirements for their system. If the system plan does not carefully delineate the computer security requirements for the proposed system, which is the usual case, the evaluator will give suggestions along with the rationale for strengthened security. The evaluator does not give specific recommendations unless asked to do so. The Developmental Systems Evaluation office will then send the assessment back to the initiator of the system plan (or the Office of Primary Concern, OPC). If the OPC for the System Acquisition Plan requests further technical help, the Development Systems Evaluation Office will work with the OPC to improve the computer security requirements in the

plan. About one half of the system plans that we evaluate return with strengthened computer security requirements.

The Developmental Systems Evaluation Office is also required, when tasked, to contribute its expertise in the selection and use of approved trusted products and new technology. At present members of this office are currently working in the development phase of several large systems including I-S/AMPE, BLACKER, WIS FORSCOM Security Monitor, and SACDIN. As an example, the DoD CSC has been tasked by the WWMCCS Information System (WIS) Joint Program Management office (JPMO) to lead the FORSCOM development effort. The FORSCOM Security Monitor project, which plans to run in a modified version of controlled mode operation, will be one of the first systems to use the Honeywell Secure Communications Processor (SCOMP) in an operational environment. The SCOMP is presently undergoing evaluation by the Evaluated Office and is a good example of the emerging new technology. Evaluators from the Developmental Systems Evaluations Office meet regularly with the contractor, users (FORSCOM), and representatives from the Department of the Army, DoD, and industry. The Developmental Systems Evaluation Office has prepared the Statement of Work and a Computer Security Requirements document. This last document has been circulated among the members of the certification working group and the contractor. The computer security requirements document will provide the contractor with the 'design-to' computer security guidelines for FORSCOM. Thus FORSCOM will represent the use of a trusted product to provide a solution to a "real-life" computer security problem.

For an operational systems evaluation, a team is assigned to do a technical evaluation of the system. This evaluation includes a detailed hardware and software vulnerability and threat analysis study. The mapping of threats and vulnerabilities can be considered to be a risk analysis. The team begins by reading the supporting system documentation, such as the System Requirements Specification and Functional Requirements Specification to gain familiarity with the system. As a part of each evaluation, the team will take classes to study the operating system for the target system, learn the system's assembly or higher-level language, and spend time talking with the prospective users of the system. The evaluation will conclude with the publishing of a technical report which details the findings of the team.

During an evaluation, the team evaluates the system's hardware, software, and configuration control against the requirements of the proposed class and any pertinent DoD directives. Because the DoD CSC is not responsible for Physical, Personnel, or COMSEC areas of computer security, we suggest that physical, personnel, and COMSEC security evaluations be done by the proper authorities. The security perimeter for the system is clearly defined, including manual and automatic trusted processes to control access to classified or sensitive data in the system. The team looks at any possible means of subverting system security on the machine, for example, as getting from user to supervisory state in an IBM MVS system, or an ordinary user guessing the Superuser password on a UNIX system and gaining superuser privileges. If the system is a candidate for the B division (Mandatory Protection), the team ensures that the labeling requirements for the Criteria are met. Test scenarios are developed and executed to test the correct functioning of the system and attempts are made to subvert the correct functioning of the system.

The Operational Systems Evaluations Office also uses "Flow Analysis" techniques (which include covert channel analysis and data flow analysis) on each system because the Criteria require the developer to search for and measure covert channels (both storage and timing) for all A and B Division candidates. We have asked several contractors (FORSCOM Security Monitor and I-S/AMPE, for example) to look for covert channels, either storage or timing, and to eliminate or neutralize such channels. The Operational Systems Evaluations Office will use the skills of Verification/ Validation experts to analyze the formal methods required for A Division candidate systems. Frequently, security vulnerabilities are discovered during the course of a system evaluation. The problems that the Center has most frequently found in older systems are:

Passwords: Sometimes displayed, frequently guessable. We are all familiar with the scenario of the user whose password is the spouse's or childrens' first names. One system we evaluated did not display the password, but did echo the number of postions in the password. In another case, we were able to guess the initial password for an account with security officer privileges. Needless to say, we gave ourselves all the privileges that we wanted.

Configuration Management: Access to system software is allowed. We found a system programmer's account, in which jobstream control file was stored. This file contained the system password for the entire complex. We could have denied service to the system by using the password to delete a system file.

Environment: Machines left unattended; emanations; level of terminal protection; external electrical connections. This is not the responsibility of the DoD CSC. We are all familiar with the "4-1-4"s Los Alamos/ Sloane-Kettering break-ins utilizing dial-up lines.

Auditing: No review of logs; insufficient information such as no individual accountability; falsifiable. One system we looked at allowed any user to look at the audit trail, change it, or delete the whole thing.

Access Control: Modifiable; can be scavanged; can be by-passed. We were able to set a field in an add-on access control package so that the number of invalid attempts would not exceed the preset limit, effectively by-passing the access control.

In the case that the evaluation team finds major security vulnerabilities, the team will report the problem and seek to have the vulnerabilities corrected. Once the flaws have been corrected, the evaluation team will retest to be sure that the system is functioning as expected. We do not, however,

work in an endless find-patch-retest mode. Finding and patching a problem will not guarantee that all security flaws have been found and eliminated.

When the operational system evaluation team has completed its testing and evaluation, a technical report will be written and sent to the organization that initiated the tasking. The report will contain an introduction, which describes the mode of operation of the system and a discussion of the suggested Criteria class. A description of how the mandatory and discretionary access controls are supposed to work for the system follows the introduction. This section will discuss identification and authentication of users; separation of users, user privileges, and user encapsulation. A section describing the hardware and software configuration control of the system is given. The next section in the report contains the relevant security requirements for the mode of operation and an item by item analysis of whether or not the system complies with the requirements. The next section contains a list of the problems found during the evaluation. A vulnerability profile is included in this section for each of the problems found. The vulnerability profile gives the type of breach (unauthorized acquisition of information, denial of service, or modification), standards area violated (identification and authentication, user isolation, audit, configuration management, and system security environment), suggested degree of vulnerability (high, medium, and low), suggested severity of the threat, a suggested cost to fix, and suggested benefit if fixed. The summary wraps up the discussion of the system. It is in this section that any environmental requirements will be given to justify the relaxing of any computer security requirments. This section also gives a judgement of how severe the security risks for the system are. The report concludes with the recommendation for or against certification. This recommendation will be part of the information used by Designated Approving Authority (DAA) to recommend or not recommend approval. The DAA has the right to ignore the technical report or to weigh the conclusion of the report in light of the system's environment. Because of the possible sensitivity, reports are not disseminated widely outside of the DoD CSC. Only the organization initiating the tasking and members of the DoD CSC may have access to the report.

The Applications Systems Evaluations Office has faced many "challenges" during the two years since we were formed. The foremost problem is the lack of environmental guidelines. A candidate class C2 system which does not support discretionary access may be perfectly fine for one application, but in another application, a similar candidate C2 system may require discretionary access control because of higher risk. Another problem concerns the "evolving system." Because the Applications Systems Evaluation Office frequently gets into the evaluation act late in the development or implementation cycle, the system may continue to change on an almost daily basis. No operating system will stay static forever. New releases will continue to be marketed for the life of the system. The evaluators must be able to work in a theater where things are not always static. The third major problem area concerns networks. The Trusted Computer Systems Evaluation Criteria do not discuss specifically how the criteria should be applied to networks. There are plans to produce network evaluation guidelines.

In summary, the Applications Systems Evaluations Office has contributed to the DoD Computer Security Center's goals by providing application support to NSA/CSS, DoD components, and to the members of the National Security Establishment. We have or are evaluating selected systems such as WIS FORSCOM Security Monitor, BLACKER, I-S/AMPE, SACDIN, and DDN. We will provide support during the acquisition process by recommending technical alternatives, utilizing evaluation standards, and giving specialized input. During the certification and accreditation process, we will provide support by providing evaluation tools, post-operational guidelines, and technical recommendations. We will keep the lines of communication open between the DoD CSC and other organizations interested in computer security by acting as the DoD focal point for computer security. The Applications Systems Evaluations Office is task-driven and works with the user when asked. For you the user, we are your interface with computer security.

PANEL SESSION - HOW DO YOU SELL BETTER COMPUTER SECURITY?

Moderator - Steven Lipner
Engineering Manager
Digital Equipment Corporation

Panel Members:

Lester Fraim - Honeywell Federal System Division
Theodore Lee - UNIVAC
Stephen Lipner - Digital Equipment Corporation

INTRODUCTION

The following panel session was intended to discuss the necessity of selling the need for computer security within an organization. The participants provided their perspectives on the need for government and industry to work together for better computer security.

The outline that follows is a list of the main points made by the three panel members. It was edited directly from the transcript of this session and was not reviewed by any of the speakers prior to publication.

KEY POINTS MADE BY SPEAKERS:

Lester Fraim

1) Computer security has to be understood by the consumer before anyone can sell it.

2) During a two-hour SCOMP presentation, three-quarters of it is spent on computer security in general while only 15 to 2Ø minutes is spent actually discussing SCOMP.

3) Different environments require different levels of security; for some people, passwords might be enough.

4) Those marketing computer security really have to understand the technical aspects so that they can be explained to all the various levels of management.

5) The establishment of the Computer Security Center has been a major thrust; government and industry have to work together to effectively use and sell the existing technology.

6) Recent hearings on Capital Hill and exposure of the 414's activities have brought the security problem to the attention of the general public.

7) There are several security products currently available: MULTICS, SCOMP, and FORSCOM Guard.

8) One of the biggest problems is selling security within an organization: convincing upper management that it's needed.

9) Security must also be sold to lower level management since most of them are used to operating in a system high mode.

1Ø) An area of disagreement is in the area of programming languages and in using structured high-level languages.

Theodore Lee

1) The Criteria has made an impact by convincing management that they should be paying serious attention to security.

2) The Center has institutionalized this need for security, and it's not likely that the Center will only be around for a few years and then disappear.

3) The Criteria has also made an effort to define computer security in a more consistent way than it was defined before.

4) Awareness of computer security vulnerability is important, but it's near the bottom of the list.

Steven Lipner

1) The effort of developing security products started with the DoD market.

2) One of the big questions is getting the technology into the marketplace.

3) One has to sell security on the outside - to consumers - then inside one's organization, then outside to the consumer again.

4) I have derived that there is an awareness of and interest in computer security in the user community.

5) The hardest things to achieve technically are the B1 and B2 levels.

6) The national security world is not a huge fraction of the market, but development costs aren't too huge, so we do get the resources.

7) People do understand that there are security exposures and that it's worth investing in development.

8) The DoD Computer Security Center and the Criteria have helped sort things out and have been a major service to the computer community.

✪U.S. GOVERNMENT PRINTING OFFICE: 1984-449-356/19031