NISTIR 5472

A Head Start on Assurance

Proceedings of an Invitational Workshop on Information Technology (IT) Assurance and Trustworthiness

March 21-23, 1994

George Washington Inn
Williamsburg, Virginia

Edited by
Marshall D. Abrams
The MITRE Corporation

and

Patricia R. Toth
National Institute of Standards and Technology

Sponsored by
Aerospace Computer Security Associates

Co-sponsored by
U.S. DEPARTMENT OF COMMERCE
Technology Administration
National Institute of Standards
and Technology
National Computer Systems Laboratory
Gaithersburg, MD 20899

NIST

NISTIR 5472

A Head Start on Assurance

Proceedings of an Invitational Workshop on Information Technology (IT) Assurance and Trustworthiness

March 21-23, 1994

George Washington Inn Williamsburg, Virginia

Edited by
Marshall D. Abrams
The MITRE Corporation

and

Patricia R. Toth
National Institute of Standards and Technology

Sponsored by
Aerospace Computer Security Associates

Co-sponsored by
U.S. DEPARTMENT OF COMMERCE
Technology Administration
National Institute of Standards
and Technology
National Computer Systems Laboratory
Gaithersburg, MD 20899

August 1994

U.S. DEPARTMENT OF COMMERCE
Ronald H. Brown, Secretary

TECHNOLOGY ADMINISTRATION
Mary L. Good, Under Secretary for Technology

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
Arati Prabhakar, Director

ABSTRACT

The purpose of the Invitational Workshop on Information Technology (IT) Assurance and Trustworthiness was to identify crucial issues on assurance in IT systems and to provide input into the development of policy guidance on determining the type and level of assurance appropriate in a given environment. The readers of these proceedings include those who handle sensitive information involving national security, privacy, commercial value, integrity, and availability.

Existing IT security policy guidance is based on computer and communications architectures of the early 1980s. Technological changes since that time mandate a review and revision of policy guidance on assurance and trustworthiness, especially since the changes encompass such technologies as distributed systems, local area networks, the worldwide Internet, policy-enforcing applications, and public key cryptography.

FOREWORD

The Aerospace Computer Security Associates (ACSA) had its genesis in the first Aerospace Computer Security Applications Conference in 1985. That conference was a success and evolved into the Annual Computer Security Applications Conference (ACSAC), which is now in its tenth year. Several years ago, "Aerospace" was dropped from the name to promote a wider range of government and commercial applications. ACSA was incorporated in 1987 as a small, non-profit association of computer security professionals who have a common goal of improving the understanding, theory, and practice of computer security. ACSA continues to be the primary sponsor of the annual conference.

In 1989, ACSA began the Distinguished Lecture Series at the annual conference. Each year an outstanding computer security professional is invited to present a lecture of current topical interest to the security community. Past Distinguished Lecture speakers have included Dorothy Denning and Willis Ware. In 1991, ACSA began issuing a Best-Paper-by-a-Student Award at the annual conference. This award is intended to encourage active student participation in the annual conference. The Distinguished Lecturer and the award winning student author receive an honorarium and all expenses paid for attending the conference.

ACSA continues to be committed to serving the security community by finding additional approaches for encouraging and facilitating dialogue and technical interchange. ACSA is always interested in suggestions from interested professionals and computer security professional organizations on achievement of these goals. In early 1994, ACSA, responding to a perceived growing need in the community, organized and sponsored the Invitational Workshop on Information Technology Assurance and Trustworthiness (IWITAT). The IWITAT is the first in what ACSA hopes will be a successful series of workshops on assurance, trustworthiness, and other topics of critical interest to security professionals. These proceedings document the activities of that workshop.

The Computer Systems Laboratory (CSL) at the National Institute of Standards and Technology (NIST) works with governments, industry, academia, and consortia to improve the efficiency and delivery of government services. CSL develops standards, guidelines, and test methods; validates products for conformance to standards; conducts research; and provides techical advice and assistance. Many interactions are accomplished working directly with industry through workshops, research and development agreements, and other cooperative arrangements.

The Computer Security Division of the CSL provides guidance and technical assistance to government and industry in the protection of unclassified automated information systems. The Computer Security Division develops, prototypes, tests and implements computer security standards and procedures to prtoect sensitive information from unauthorized access or modicification. Areas of cooperation with industry include risk management; open systems; LAN security; security architectures; systems integration; and public and private key cryptographic techniques as applied to electronic data interchange (EDI), electronic funds transfer, and electronic mail.

PREFACE

ABOUT THIS INTERNAL REPORT

This Internal Report contains many more questions than answers, accurately reflecting the current state of knowledge and practice. It is, however, a significant step toward developing an agenda for future work in IT assurance and trustworthiness.

The plenary and working sessions are each documented in a separate section of this Internal Report. The reader is assumed to be knowledgeable about information security and familiar with the *Trusted Computer System Evaluation Criteria* (TCSEC) (DOD, 1985) as well as more contemporary efforts, such as the *Canadian Criteria* (Canadian System Security Center, 1993), the European *Information Technology Security Evaluation Criteria* (ITSEC) (Commission of the European Communities, 1991), and the United States *Federal Criteria (FC) for Information Technology Security* (National Institute of Standards and Technology and National Security Agency, 1992).

For background reading, we suggest *Computers at Risk* [National Research Council, 1991]. Also, publication of *Redefining Security* (Joint Security Commission, 1994) shortly before the workshop provided an authoritative source for many commonly held opinions. Specifically the following chapters cite directly relevant background to this Internal Report: Chapter 1, "Approaching the Next Century"; Chapter 8, "Information System Security"; and Chapter 11, "A Security Architecture for the Future."

ACKNOWLEDGMENTS

We wish to acknowledge the assistance of Heidi Bass of the George Washington Inn for her support during the post-workshop editing session. C. Dawn Gibson and Carol R. Oakes of The MITRE Corporation helped transform independent contributions into a coherent proceedings.

TABLE OF CONTENTS

LIST OF FIGURES

EXECUTIVE OVERVIEW

INTRODUCTION

The twofold purpose of the Invitational Workshop on Information Technology (IT) Assurance and Trustworthiness was to identify crucial issues on assurance in IT systems and to provide input into the development of policy guidance for determining the type and level of assurance appropriate in a given environment and talk about also guiding future directions in this area. The workshop participants defined *assurance,* as applied to IT products and systems, as the degree of confidence that security needs are satisfied.

Existing IT security policy guidance is based on computer and communications architectures of the early 1980s. Technological changes since that time mandate a review and revision of policy guidance on assurance and trustworthiness. These changes encompass such technologies as distributed systems, local area networks, the worldwide Internet, policy-enforcing applications, and public key cryptography.

There is a growing consensus that no one technique can provide comprehensive adequate assurance. Established approaches need to be re-examined and compared to newer ideas. Major issues and concerns include the following:

- How architecture contributes to assurance
  - The balance of assurance between operating systems and applications
  - The management of information security for subscribers in a worldwide information infrastructure
  - The use of larger, more heterogeneous computing environments
- The growing requirement to enforce policies other than confidentiality, such as the following:
  - Integrity and availability (primarily)
  - Non-repudiation or anonymity (occasionally)
- The relationship between process and assurance

Since no metrics exist for determining the effectiveness of assurance techniques, it is very difficult to compare the various techniques. Nevertheless, (GAO, 1994) and (JSC, 1994) recommend approaches for improved cost-effectiveness.

WORKING SESSIONS

The workshop was structured into six working sessions, each of which is summarized below.

Tradeoffs

Assurance effort for any system should be balanced based on perceived risk and cost. Application assurance, which has generally been underemphasized in the past, should be addressed along with product assurance. There is no single uniform approach to assurance that will satisfy all kinds of system applications. To support a balanced approach, assurance arguments should be assembled from a set of system building blocks. Concepts of system composition and integration should allow the assurance analysis to be tailored to specific user requirements. Assurance evidence should be carefully packaged to best support enterprise decision-makers during the security tradeoff process.

Pedigree

The Pedigree session focused on the acceptability of assurance evidence based on the identity of the creators of that evidence. That is, participants discussed assurance evidence based on the "who" rather than the "what" or "how." The term "credentials" would also have been a good name for this session. The issues can be logically grouped into one of the following eight categories:

- Applicability
- What is "pedigree"?
- Assurance based on an individual
- Assurance based on an organization
- Measurements/metrics
- Enforcement and liability
- Usefulness

- Aggregation concerns

Drawing on the similarities and differences among IT security professionals and other professionals such as engineers, certified public accountants, and lawyers, much of the general discussion addressed the value/drawbacks of formalizing an informal aspect of assurance.

Security Architecture and Applications

This session aspired to identify opportunities for managing information security for subscribers in a worldwide information infrastructure, to learn how architecture contributes to system assurance, and to understand how application-specific requirements can be addressed in this context. We observed a growing requirement to enforce policies other than confidentiality (e.g., integrity, availability, non-repudiation, or anonymity).

Achieving security solutions that are applicable in a larger, more heterogeneous computing environment requires a shift from the current paradigm of developing security for each individual system toward recognition that security is a global property that must be addressed throughout the computing environment. Moreover, each component or system within the environment has a role to play in the protection of information. Conventional security approaches can ensure that these systems or components comply with their defined role. Some of these approaches were discussed in detail, including the development of trusted applications and products that support the enforcement of different policies.

Process

Two fundamental issues for this session were (1) whether improved and uniform processes for development and evaluation will lead to higher quality and more predictable evidence and, hence, better, faster, less expensive assurance and (2) to what degree can assurance about process contribute to assurance about products and systems.

The following major topics emerged: assurance about a process; the relationship of process to system/product assurance; and process improvement. Fundamental issues regarding process and assurance and their interrelationships include reliance on process, particularly the development process, as a major component of system and product assurance; measurement of process quality and adherence as a basis for assurance; and understanding assurance in terms of concepts such as correctness, effectiveness, and workmanship.

Many interrelated processes exist today; these processes may be formally stated or conducted in default. While a single integrated process sounds attractive, it most likely would be too complex, too high-level, and ineffective.

Metrics and Testing

The session identified five different types of assurance and, for each type, relevant assurance techniques. This session discussed what assurance is, how it can be measured, both qualitatively and quantitatively, and how testing, including automated testing, fits in to assurance. Having identified five different types of assurance, the group identified relevant assurance techniques for each type. The five identified types were: policy assurance, design-effectiveness assurance, system-correctness assurance, evaluation assurance, and ensuring balance among the first four types.

This session's main conclusion regarding assurance measurement was that we have not been collecting the types of data needed for reliable measurements.

Risk Management

We cannot pretend that by implementing certain security measures we can mitigate all security risk. This paradigm shift from risk avoidance to risk management (i.e., risk tolerance) brings to light the necessity of dealing with the unimaginable. It is also necessary to consider incomparables such as system security, human safety, and personal career. Viewing risk assessment from the standpoint of assets and threats is necessary but not sufficient.

There are several kinds of risks to which systems are subjected, including technical, schedule, cost, security, and safety. Security risks need to be considered in the context of overall risks. Satisfaction of all objectives and avoidance of all risks are generally impossible to achieve because the objectives or techniques used to achieve these objectives are often in conflict.

The members of this session felt that much of current risk management methodology is an attempt to use the scientific method for a problem that has not been reduced to science.

Fundamental and very hard questions need to be answered to make any progress on risk management: What are the security requirements, including assurance? In what ways do we risk not meeting those requirements? How much are we willing to spend to mitigate those risks and to what degree? What should be the government's role in helping to protect information held by private citizens and institutions? How can government technology be provided to the private sector for the protection of sensitive unclassified information? Will the private sector accept that technology?

CLOSING

The general consensus was that a forum to discuss the issues of IT security assurance was of great use to the community. In particular, this workshop determined that assurance is still a somewhat nebulous subject. There are many questions that need to be explored. It is still difficult to define precisely what is meant by assurance, and the definition varies from person to person and enterprise to enterprise. The questions of how to gain assurance, how to convey assurance results, and how to use assurance all need further study. It was generally felt that just the identification of these questions for further study made the workshop a useful exercise.

There was some sentiment that these subjects need to be pushed back out into the community for actual resolution. The security community appears to have made little progress in truly understanding the issues at hand, and there is little hope for the immediate future. While there appeared to be much agreement on what had been done incorrectly in the past, there appeared to be little consensus on how to proceed. One thing appears clear; in order to improve the security of our information and resources we must:

- Respond in a timely manner to rapidly changing technology and threat environment
- Becomes more proactive and more in touch with the real user needs and expectations
- Does a better job of developing security awareness in the user community (to ensure security is built in *and* maintained during operation of the system).

SECTION 1

INTRODUCTION

Before entrusting valuable information assets to an IT system and placing an organization in a position of depending on the confidentiality, integrity, and availability of these assets, responsible management must be convinced that the IT system is sufficiently trustworthy to meet the needs of its operational environment.

Work is under way to produce new national and international criteria for IT security. These emerging criteria ascribe to the paradigm of analyzing a given environment to identify risk, threat, and vulnerability; determining applicable legislation, policy, and custom; and selecting administrative, physical, and technical countermeasures that reduce the residual risk to an acceptable level. These criteria describe IT security functionality and approaches for assurance, but they do not include guidance on how to determine the appropriate and necessary assurances for a particular environment.

By way of contrast, the TCSEC assumed the presence of a human adversary who attempts to cause the IT system to behave in a way contrary to that for which it was intended. While this assumption has had a dominating influence on assurance for IT products designed for the military, it is clearly not accurate for all operational environments.

1.1 PURPOSE OF THIS WORKSHOP

The purpose of this workshop was to document the perceived state of practice and stimulate new ideas concerning assurance in security-relevant IT systems. Input from people in the field who must make decisions about using IT was especially sought. Publishing this information is designed to provide input into the development of security policy guidance on determining the type and level of assurance appropriate in a given environment. Practically all existing security policy guidance is based on the Yellow Books, published in 1985 (National Computer Security Center, 1985). This guidance was based on computer and communications architectures of the 1980s. This workshop addressed questions of how policy guidance on assurance and trustworthiness needs to be revised in order to stay consistent with technological changes, especially those brought about by distributed systems and related technologies such as local area networks, the worldwide Internet, policy-enforcing applications, and public key cryptography.

Security policy guidance combines aspects of technology assessment, risk analysis, and cost-effectiveness. Trade-offs between academic and cost-effectiveness considerations must be made. Answers must be given in the face of inadequate information and technical uncertainty.

The mission of the workshop was to identify the crucial issues and to make recommendations for future direction in this area. Readers of these proceedings include those who handle sensitive information involving national security, privacy, commercial value, integrity, and availability.

Participants submitted position papers expressing technical or policy views, and these position papers were used to identify working session topics. All accepted position papers were distributed as anonymous to all participants by e-mail in advance of the workshop. It was felt that anonymity would better serve the objective that position papers should stimulate discussion and not necessarily represent final positions. For the same reason, the position papers are not included in this publication.

1.2 PRELIMINARY LIST OF ISSUES

A preliminary list of issues, presented below, was offered to focus and stimulate the position papers. Not all of these issues were discussed. In fact, the entire workshop identified many more problems than it solved. These issues were used to develop the working session topics listed in Section 1.3.

- Assurance is not a one-dimensional quantity, but a vector with many components, one for each perceived threat in the user environment.
- Threats need to be more cost-effectively aligned with appropriate countermeasures, especially in environments where the threat changes dynamically.
- Precedent may not be the best indicator of what assurances are acceptable.
- Practical considerations of cost-effectiveness strongly suggest that pre-deployment product assurance should be balanced with operational assurances in the area of ongoing system operations and maintenance.
- There is a danger in becoming too comfortable with the alleged intractability of perfect security. We are accepting the current crop of half-measures as the best attainable.

- Kinds of guidance that are possible and appropriate need to be identified.

- That guidance can take various forms:

  - Simple formula like Yellow Book risk index
  - One hundred possible scenarios for best match

- The benefits of high-assurance techniques such as formal methods need to be examined

  - For cost-effectiveness
  - Comparison to original expectations.

- Sufficient assurance in some environments can be provided by several methods:

  - Conforming with process standard such as International Organization for Standardization (ISO) (ISO, 1987) 9000.
  - Employing good software engineering practice
  - Providing limited warranties for repair of security flaws
  - Using capability maturity models developed by the Software Engineering Institute at Carnegie-Mellon University.

- Direct characterization of product strength is needed, in terms of the difficulty of exploiting the flaws in the product. One would say that the product is suitable for a particular use if its security controls are harder to defeat than some pre-determined threshold.

1.3 WORKSHOP ORGANIZATION

There was an opening plenary session, followed by six working sessions divided into two parallel tracks. Each working session had a moderator and recorder, listed below, who produced these proceedings. All of the attendees were given an opportunity to copy edit these proceedings.

The moderator for each working session started the activities by summarizing the issues and suggestions. The moderator's remarks were based, in part, on the position papers submitted. Although the position papers were not presented at the workshop, the authors had the opportunity to remain anonymous, to keep to the position they wrote, or to change their minds. Some, but not all, of the authors identified themselves.

The working sessions, moderators, and recorders were as follows:

Security Assurance Tradeoffs
Moderator: Bret Hartman
Recorder: Lynne Ambuel

Pedigree
Moderator: Deb Campbell
Recorder: Pat Toth

Security Architecture and Applications
Moderator: Judy Froscher
Recorder: Jay Kahn

Process
Moderator: Joel Sachs
Recorder: Caralyn Wichers

Metrics and Testing
Moderator: Jim Williams
Recorder: Caralyn Wichers

Risk Management
Moderator: Marshall Abrams
Recorder: Lynne Ambuel

1.4 TERMINOLOGY

Participants agreed that the pronoun *we* is often overused or ambiguous. Usually it refers to the set of people working on information security issues, sometimes called the *information security community*. Sometimes it referred to the technical subset of the information security community, excluding the managers. Sometimes it is the regal we. Occasionally, it referred to us—the workshop participants. We hope that no one is unnecessarily confused by this usage. Any confusion in this publication probably reflects how well we understand each other.

1.5 DOCUMENT ORGANIZATION

As stated previously, the remaining sections of this document correspond with the plenary and working sessions of the workshop: Section 2, "Opening Plenary"; Section 3, "Security Assurance Tradeoffs"; Section 4, "Pedigree"; Section 5, "Security Architecture and Applications"; Section 6, "Process"; Section 7, "Metrics and Testing"; Section 8, "Risk Management." The document concludes with closing statements cited in Section 9 and an appendix describing security services applied to the THETA system.

SECTION 2

THE OPENING PLENARY

The opening plenary session began by defining the objectives of the workshop: provide input to policy guidance in the face of inadequate information and technical uncertainty; identify the type and level of assurance appropriate in a given environment; combine and trade-off aspects of technology assessment, risk analysis, and cost-effectiveness, recognizing that we may not be able to afford academic completeness; and identify crucial issues and make recommendations. Participants were asked to focus on assurance and resist the temptation to solve all information security problems.

Several sets of extracts from *Redefining Security* (Joint Security Commission [JSC], 1994) set the stage for discussion. While the report address the situation in the United States (U.S.), it is probably applicable to other countries as well. The motivational observations included:

- Protecting the confidentiality, integrity, and availability of the nation's information systems and information assets—both public and private—must be among our highest national priorities.
- IT is evolving at a faster rate than information systems security technology.
- A systems approach is necessary in making decisions about the application of security countermeasures.
- Countermeasures are frequently out of balance with the threat, often based on worst-case scenarios rather than realistic assessments of threats and vulnerabilities.
- Security is a service that should be based on an integrated assessment of threat, vulnerability, and customer needs.
- Security is a balance between opposing equities.

The JSC observations about threats to information and information security included:

- Networks are recognized as a battlefield of the future.
- An attack on unprotected civilian infrastructures could be disastrous.

- Foreign intelligence services, including those of some of our "allies," are known to target U.S. information systems .

- Computer viruses, other malicious software, and hackers are increasingly common and dangerous.

- Hiding information about security flaws from ourselves doesn't help.

- Eighty-five percent of computer crime is committed by insiders with validated access.

Observations about failed strategies served to remind the participants that business as usual is unacceptable:

- Encouraging the private sector to design, develop, and manufacture products at their own expense against government promise to require their use: the government did not follow through and buy the products.

- Research has focused on classified information to the detriment of protecting unclassified information and infrastructure.

The JSC-recommended strategies directed the focus to constructive criticism:

- Promoting understanding in the private sector that it is less expensive to protect information assets with affordable technology than with insurance should result in availability of moderate-assurance security products.

- Government funding is necessary to promote development of high-assurance products.

- It would be reasonable to allocate five to ten percent of total development and operational cost to ensure availability, confidentiality, and integrity.

- Research and development (R&D) should be coordinated and focused on products for protection of classified and unclassified networks and systems.

- Infrastructure security management should be given more attention.

SECTION 3

SECURITY ASSURANCE TRADEOFFS

Bret Hartman, Moderator
Lynne Ambuel, Recorder

3.1 INTRODUCTION

This session addressed security assurance tradeoffs. Security assurance tradeoff decisions occur in many contexts, such as assurance benefit versus cost, the relationship of assurance to system functionality, and the requirements on assurance imposed by the value of information stored in the system. The group discussed a variety of tradeoff issues as well as the current views of the security community. Tradeoffs issues were a fundamental theme of the workshop—the group discussions cut across many of the other sessions.

The position papers discussed during this session were representative of the growing opinion that the *Department of Defense Trusted Computer System Evaluation Criteria* (TCSEC) (Department of Defense [DOD], 1985) does not necessarily have the appropriate emphasis on assurance. The TCSEC focuses all efforts on the trusted computing base (TCB), which in the traditional view is the collection of hardware and software that enforces the underlying system security policy. By enforcing the security policy, the TCB thus ensures that untrusted non-TCB software may safely access sensitive information without danger of compromise.

Experience has shown, however, that there are several problems with this approach. Traditional high-assurance systems (e.g., B3 or A1) are difficult to use because of limited functionality and the potential impact on performance. It remains to be determined whether it is practical to produce high assurance and high functionality products. In addition, the assurance evidence required is difficult and time-consuming to produce and evaluate.

Furthermore, a traditional TCB may not adequately address the security requirements in applications. Applications include general-purpose systems such as Database Management Systems (DBMS) and financial management systems, highly focused systems such as process control, and in-between systems that focus on a broad market segment such as a military message system.

Applications frequently have different security tradeoffs than the underlying TCB. The narrow security policy of the TCB may not be sufficient to protect against many application-specific

security requirements in areas such as accountability, availability, integrity, and the prevention of information leakage through known covert channels in the TCB.

The group discussed possible solutions to the limitations of the TCSEC approach to TCB assurance. One potential direction is to extend the system security perimeter to include both the TCB and the Controlled Application Set (CAS). The CAS is the set of applications that have access to sensitive information and thus are subject to additional constraints beyond those enforced by the TCB. Identifying the CAS is consistent with security measures in practice today, where certain critical applications are carefully controlled and analyzed.

Another direction is to recognize that certain security functions beyond those typically implemented within the TCB can help augment application security. For example, containment mechanisms can limit damage caused by rogue applications. Application-specific security checks within the application may also contribute significantly to overall system assurance.

Finally, the notion of balanced assurance was a common discussion theme during the session. Balanced assurance promotes the use of assurance techniques appropriate to the level of risk in system components. Based on the level of risk for specific components, different assurance techniques would be used as appropriate. For example, if discretionary access enforcement were considered a lower risk than mandatory access control for some system application, then mandatory access control mechanisms would be subject to much greater scrutiny.

In order to make any security assurance approach feasible, we must recognize that it is not possible to eliminate the risk of a security compromise. Security assurance must always be a balance between cost and perceived risk. The primary assurance tradeoff involves one central issue: the balance of assurance cost versus the resulting security benefit.

3.2 MAIN CONCEPTS DISCUSSED

In order to focus the discussion of assurance tradeoffs, the group discussed four concepts, presented below as questions:

- What is wrong with the TCSEC emphasis on TCB assurance?
- Why is security of a TCB different from security of an application?
- Is it possible to quantify assurance versus vulnerability tradeoffs?

- Who (e.g., policy makers, vendors, application developers, accreditors) should make tradeoff decisions?

Each of these questions is addressed below.

3.3 WHAT IS WRONG WITH THE TCSEC EMPHASIS ON TCB ASSURANCE?

The TCSEC emphasis on the TCB is inflexible—the TCSEC assumes a pre-defined set of threats (e.g., Trojan horse attacks) that are not necessarily relevant in all systems. Based on these threats, the TCSEC asserts that the TCB is the totality of required security mechanisms. In most systems, the TCB is the subset of the operating system that enforces access control, especially Bell-LaPadula (Bell-LaPadula, 1975) properties. Although the TCB has a significant role in enforcing system security, practice has shown that applications also have a large role. For this reason, it is important to examine how assurance applies to applications.

In the non-DOD commercial world, assurance has a different emphasis. TCSEC assurance is usually too demanding for commercial applications. *Assurance* appears to have a bad connotation for many customers, indicating a product that is expensive and does not necessarily provide good performance. Despite the apprehension about high-assurance systems, customers still want the same result: they want to be sure that a product works as it is intended.

The provision of product warranties appears to be a growing trend for assurance in the commercial world. Although the most common approach to commercial software development is perceived to have been to release unreliable code first and then fix bugs as they are discovered, this approach is becoming less common. X/Open, for example, is developing a branding process that entails the endorsement of software backed by a clearly defined vendor commitment to a defined standard of quality. When a vendor makes a public pronouncement that its UNIX system conforms to X/Open's branding scheme, it is committing to maintain this standard. If it is shown that a product does not actually meet X/Open requirements and the vendor does not correct the problem in a timely fashion, then that vendor may lose the right to use the X/Open Logo. When and if this becomes public knowledge the vendor may suffer embarrassment and lose market credibility and market share.

The concept of a system containing "no obvious flaws" would be desirable as a basic security assurance requirement. If a standard list of known flaws were published and frequently updated, products could be tested against the list. This notion is similar to the current approach taken by virus-detection software. Although it is well known that assurance based solely on existing penetration attacks is inherently limited, this simple assurance approach

would be a significant improvement over the current state of practice in most commercial systems.

3.4 WHY IS SECURITY OF THE OPERATING SYSTEM TCB DIFFERENT FROM SECURITY OF AN APPLICATION?

In is generally accepted that the operating system cannot address all aspects of application-specific security because it is not possible to provide all security in an application-independent manner. There is some debate whether the security-relevant part of an application should be considered part of the TCB. Although an operating system TCB provides the underlying basis (e.g., identification and authentication) for building application security properties, it has traditionally emphasized enforcement of confidentiality. Applications tend to support other security properties, such as integrity and availability, that are specific to the problem domain. Tradeoffs of security properties (e.g., integrity versus confidentiality) must be based on operational requirements rather than a priori constraints defined in the TCSEC.

The group agreed that the focus of system security must shift. Too much emphasis has been placed on assurance of products, and too little has been placed on assurance of systems. We need to spend more time and effort building and analyzing operational systems; product evaluation is only the first step of this process. Furthermore, we need to emphasize upgrading and maintaining the level of assurance once a system becomes operational. Assurance maintenance continues to be a neglected area for both products and systems.

Basic building blocks that define subsystem abstractions are a critical aspect for developing assurance. Rather than a monolithic assurance approach, wherein the developer must follow a predefined formula, assurance in the form of building blocks allows developers and users to tailor assurance to their particular system. For example, military systems that primarily require confidentiality have traditionally concentrated on assurance of the underlying operating system, while commercial systems requiring integrity may address application software much more heavily. As discussed in Section 5, application security is becoming increasingly important for confidentiality policies as well. It may be appropriate to concentrate assurance on the areas with the most perceived risk.

The TCSEC approach of relying on the operating system to provide integrity to all security-relevant functions needs to be re-examined. Its validity limits need to be probed and alternative bases, if any, need to be explored.

In order to define system building blocks, we need assurance guidance for interconnecting subsystems, applications, and products. Further research is still required in this area—the

concept of system composability, although defined for limited domains, needs to be developed further. Support for distributed systems, which are gaining widespread use, is particularly important. We also need to refine the concept of product integration. The fundamental issue is one of systems engineering: how to integrate subsystems into a larger system while preserving assurance.

3.5 IS IT POSSIBLE TO QUANTIFY ASSURANCE VERSUS VULNERABILITY TRADEOFFS?

Many issues must be considered when performing assurance tradeoff analyses. This difficult task becomes even more complicated because many of the decisions are highly subjective. If quantitative measures to aid in tradeoff analyses could be developed, complexity of the tradeoff decision might be reduced.

The group generally agreed that it was easier to quantify assurance tradeoffs by focusing on security vulnerabilities rather than security risks. Doing this may create a gap between quantification efforts and the risk containment and reduction goals identified in Sections 1, 3.4, 5, and 7.4, however. Current risk-analysis techniques are largely subjective, and are based on the judgment of expected threats to system security in a given environment rather than on actual experience. Objective risk-analysis techniques are discussed briefly in Sections 8.1 and 8.7. Security vulnerabilities can be largely assessed using purely technical criteria—while a particular vulnerability either exists or does not exist in a system, ease of exploitation may vary considerably. Because vulnerabilities can be identified objectively, they provide an attractive means for trading off against assurance. Assurance considerations in assurance/vulnerability tradeoffs are correctness of the mechanism/product/system, and the effectiveness of the mechanism in protecting against the perceived threat, often quantified as the level of effort required to subvert the assets being protected. Additional relevant kinds of assurance are identified in Section 7.5.

It was generally accepted that the assurance paradigm presented in the TCSEC has been insufficient to address tradeoff decisions between types of assurance and vulnerability. The TCSEC prescribes specific assurance measures for avoiding vulnerabilities. However, the TCSEC does not consider many of the tradeoff decisions now being made in practice, such as tradeoffs between assurance measures and requirements on usability and performance. This limitation has often resulted in avoiding awkward trusted configurations during peacetime operations, hoping that the security features will work properly during a crisis.

Making tradeoff decisions among mechanisms that perform a specific security function is a relatively straightforward task. Unfortunately, this aspect is only a small portion of the

tradeoffs to be made. Cost, functionality, mechanisms, and assurances are all part of the decision. It is often difficult to make tradeoffs among these concerns because they are incomparable. However, these inter-disciplinary tradeoffs are often the ones that have the most impact on the product/system.

The complexity of the tradeoff decision changes depending on the subsystem being addressed—the larger the scope, the more complicated the decision. Tradeoffs made at the component (product) design level can be very straightforward. System-level tradeoffs can be more complicated because all of the composing products/systems must be considered. There are also tradeoff decisions made at the enterprise level that take into account the non-technical factors (e.g., social, legal, way of doing business). These tradeoff activities are not independent and are highly influenced by each other. An enterprise tradeoff decision is likely to influence the tradeoff decision made in the system and therefore in the design of the components of the system.

The group discussed possible metrics for a decision-maker to use when deciding on how much assurance should be included in a given product/system. The classic decision factor has been provided to the decision-maker by the assurance levels enumerated in the TCSEC and the companion Yellow Book. These prescribed levels have proven to be inadequate for many decision-makers because they do not take cost into consideration.

A member of the group suggested that assurance cost as a percentage of development cost may be a good tradeoff metric. The group decided that this could be one useful tradeoff, but did not consider several important factors. It is difficult to measure development cost because life-cycle costs are often open ended. Also, the group felt that the value of the information protected needed to be taken into consideration—a small, relatively inexpensive component that performs a crucial function may need considerably more assurance, as a percentage of development cost, than a large multipurpose system.

The problem with adding information value into the tradeoff discussion is that the value of information is difficult to ascertain. An enterprise would need to determine the value of its information for both the enterprise and for its adversaries. In some circumstances, a value cannot be placed on information (e.g., loss of life, enterprise survival). This complicates the quantification further. In the end, it was decided that the determination of information value will always have a subjective component. Although it is a factor in considering tradeoffs against assurance cost, it cannot be relied upon as the sole factor.

3.6 WHO (E.G., POLICY MAKERS, VENDORS, APPLICATION DEVELOPERS, ACCREDITORS) SHOULD MAKE TRADEOFF DECISIONS?

The group agreed that the trend for the assurance tradeoff decision-makers is moving away from policy makers and toward developers and system enterprises. While the TCSEC promotes a fixed set of tradeoff options embodied in the criteria assurance levels, most systems require further assurance decisions based on the specific needs of their application.

To evaluate assurance tradeoffs, it is necessary to recognize the role of an organization's "risk-taker". This authority is the member of the enterprise who has the responsibility for deciding the level of acceptable security risk that is appropriate for system installations. The risk-taker makes primarily political and operational management decisions based on the enterprise goals and the perceived threat environment.

Security assurance documentation is the principal technical information supplied to the risk-taker. Those people providing the security assurance analysis have the responsibility to supply to the risk-taker with the most complete and accurate assessment possible. However, this information, which will be highly technical, may not be in a form easily understood by the risk-taker. Assurance documentation must be packaged to define overall assurance in the best form possible to help risk-takers do their job.

High-level managerial decisions that are based on complex technical rationale can be difficult to formulate. The documentation must discuss alternatives in a form so that the risk-taker can clearly see possible tradeoffs. One approach to support tradeoff decisions is to provide a set of security questions for the risk-taker to consider. If the risk-taker is satisfied that the answers to the questions sufficiently address the enterprise security goals, then the risk-taker has a basis for making a decision. In this manner, security is driven primarily by the system procurement authority and program manager rather than by product developers.

3.7 CONCLUSIONS

Participants in this session discussed a wide variety of topics related to security assurance tradeoffs. In general, the group appeared to reach consensus on the current state of security assurance tradeoffs as well as directions for future investigation. The group felt that application assurance has received too little emphasis in the past, and it should receive more. The assurance effort must be balanced for any system based on the perceived system risk. There is no single uniform approach to providing assurance that will satisfy all kinds of system applications. To support a balanced approach, assurance arguments should be assembled from a set of system building blocks. Concepts of system composition and integration should allow

the assurance analysis to be tailored to specific user requirements. Models of product, system, and enterprise tradeoffs should be used to help identify the levels of tradeoff decision-making. Finally, the group recognized the role of the "risk-taker" as the decision-maker of enterprise security tradeoffs, and advocated that assurance evidence should be packaged to provide the best support to the tradeoff process.

SECTION 4

PEDIGREE

Deb Campbell, Moderator
Pat Toth, Recorder

4.1 INTRODUCTION

As presented in one of the submitted papers, *pedigree* suggests a method to determine the acceptability of evidence based on the identity of the creators of that evidence. Pedigree may be based on an individual's identity or on the organization that produced the evidence.

The goal of this session was to articulate the pertinent questions and identify the issues pertaining to pedigree. The sessions began by highlighting some of the main points of the submitted papers as a means of stimulating ideas and discussion. The main points as presented were the following:

- Where evidence of assurance is not suitable for being reused or being validated, the pedigree of the evidence can be used as a surrogate.
- The acceptance of a checklist instead of the evidence itself is essentially the reuse of evidence by the acceptance of its pedigree.
- An accreditor might accept the pedigree of the certifier and use the associated report as the primary basis for the accreditation decision. (Additionally, the challenge was posed that if the accreditor bases the decision on pedigree of certifier, was the report even necessary?)
- The practice of accepting evidence based on pedigrees will lead to islands of trust.
- Evidence with a weak pedigree could/would/should be subjected to more scrutiny.
- A pedigree may derive from the tools used.
- Most users make the mistake of narrowly basing their judgments on personal experiences. For example:

- If they know and trust product developers, the process the developer follows is less important.

- If they know and do not trust the developers, no process will convince them that the developer's product is of high quality.

- Logically, it follows that they might trust developers they do not know as long as the developers have name recognition.

4.2 ISSUES ASSOCIATED WITH PEDIGREE

The introductory period was followed by a brainstorming exercise among the eleven session participants. Each idea was recorded and later organized into groups of issues based on the natural relationships between each issue[1] .

The groups of issues provided below are the initial result of the session. Due to time constraints, only minimal sanity checks were made to determine if all issues were correctly grouped or if other relevant issues were missing.

4.2.1 What Is A Pedigree?

- Confidence can be based on tools, people, and organizations. How can they be balanced? Does one weigh off more than another?

- How do you apply pedigree to an unknown entity?

- How is a pedigree established? Can a new company gain a pedigree by hiring a consultant with an established pedigree?

---

[1] The groups shown below were developed using one of the Juran Management and Planning Tools known as an Affinity Diagram. Each issue identified during the brainstorming session was written on a post-it note. All post-it notes were then randomly placed on the wall and then moved into groups. During the process, no one spoke, and everyone worked simultaneously. The process continued until everyone was satisfied with the groups. When one issue appeared to be appropriate in two or more groups, duplicate post-it notes were made.

- Maybe the pedigree concept is misnamed and should be called *credentials*. Credentials has an updating connotation to it, rather than a birthright which pedigree implies.

- Should pedigree provide assurance based on what you have done in the past, as opposed to what you are currently doing? Once a pedigree is established, how long is it valid without being updated?

- Should pedigrees be multidimensional and not oversimplified? That is, a simple one word label for a pedigree may not convey enough information to be useful.

- A pedigree can be thought of as an integrity label. However, to be useful, must pedigrees be considered in a broader context?

4.2.2 Applicability

- For which qualities of a system or product is pedigree a useful measure (e.g., correctness, effectiveness, reliability, workmanship)?

- Is it agreed that pedigree makes a difference based on roles (e.g., accreditor, developer, security engineer, integrator, certifier, criteria writer, evaluator, profiler, user)?

- Can pedigree be extended to a tool or is it limited to individuals and/or organizations?

- What does pedigree buy me? What evidence can I waive if I have pedigree? Do I need to provide more or less evidence based on a positive, nonexistent, or negative pedigree?

- How does pedigree feed into assurance?

- What is the relationship among effectiveness, correctness, and pedigree?

- Can pedigree and criteria be viewed as opposites? Criteria in some cases may be viewed as overriding pedigree and in other cases there may not be a conflict. Perhaps criteria are not necessary.

- Does a pedigree have coattails? Is pedigree extendable? Is pedigree applicable across all products made by developer?

4.2.3 Individual Pedigree

- Should we be skeptical of individual pedigrees? Loyalty to your company may be an overriding factor. Individuals may be stifled due to loyalty to the company or organization.

- Should we decide to read something or not depending on who the author is?

- What is the importance of your personal knowledge/trust of people by name?

- What is the importance of people to the organization's pedigree? If a person leaves a company, does the trust level go down unless an "equally known and respected" person replaces him/her?

- What are the elements of an individual pedigree, such as knowledge, training, and precedent?

4.2.4 Organizational Pedigree

- Why are we skeptical of organizational pedigrees? If pedigree is associated with too large an organization, it may lose meaning. If worldwide conglomerate XXX has a pedigree, does every division carry that same pedigree or should the pedigree be limited to divisions?

- What is the importance of people to the organization's pedigree? If a person leaves a company, does the trust level go down unless an "equally known and respected" person replaces him/her?

- What is the level of granularity of pedigree associated with organization.

- How do small companies develop a pedigree?

- What is the difference/similarity between individual versus organizational pedigrees?

4.2.5 Measurements

- How can we collect security cost data?

- Can regression analysis be used to establish risk?
- Who decides what the standards are for pedigree? Who selects the judges?
- Should there be a licensing requirement for pedigrees as in other professions, such as certified public accountants, doctors, or lawyers? This topic also raises the issue of liability.
- How should we institutionalize the process for evaluating pedigrees? Should the process be institutionalized?

4.2.6 Enforcement

- Are individuals/organizations true to their pedigree? How do you know if the process is followed correctly once the pedigree has been established?
- Can you regain a lost pedigree?
- What is the liability of a pedigree?
- Pedigree is not a one-time stamp, so what elements do you re-evaluate?
- How do I protect my interest against a company with a pedigree?

4.2.7 Usefulness

- How do we capture what an individual or an organization has learned during the process?
- Should we give the "edge" to new products? What should be the balance to pedigree and innovation? Magazine reviews always tend to favor the newer, slicker products.
- What are the ramifications of "buying" based on pedigree?
- Are pedigrees useful for high-assurance systems? Or should they be limited to low-assurance systems?
- Is pedigree applicable for all products?
- Is there anything you can rely on except pedigree for legacy systems?

- Regarding networks of trust, if a trusts b and b trusts c, should a trust c?

4.2.8 Aggregation

- If we have a large system and a large number of people working on it with individual pedigrees at various levels, what is the pedigree associated with the system?

- Does a pedigree associated with a system change during the life cycle of that system? If the developer had a high pedigree but the integrator had a low pedigree, what is the pedigree of the system?

- How do we determine pedigrees on components of systems? What are the rules of composition for pedigrees?

- If you have a strong pedigree early in the process, do you need a strong pedigree late in the process? Conversely, if you have a weak pedigree early in the process, do you need a strong pedigree late in the process? This is a certification problem.

4.3 CONCLUSIONS

The session concluded with a discussion that focused on several key issues, listed below.

4.3.1 Terminology

One must be careful when introducing a new term, such as *pedigree*. The term itself may project a meaning that may or may not accurately reflect the intended concept. It is critical that the corresponding concept be thoroughly explored and the terminology match the intended meaning. This workshop allowed participants to delve into this concept in great detail. As a result, it become evident that pedigree, although used almost exclusively in this session, is not the best term to capture the intended concept. The dictionary definition of pedigree is a line of ancestors; lineage. This infers a birthright that is clearly not intended.

An alternate term suggested was *credentials*. (A dictionary definition of credentials is (1) that which entitles one to confidence, credit, or authority; or (2) evidence or testimonials attesting to one's right to credit, confidence, or authority.) Although the group did not reach a consensus during the session to adopt the term *credentials* in lieu of *pedigree*, it was felt that a term such as *credentials* was probably more appropriate for conveying the intended concept.

4.3.2 Categorizing

The concept of grouping pedigree or credentials into three categories based on roles of the individual/organization was introduced. Earlier in the brainstorming exercise, it was suggested that this concept could apply the same or differently depending upon the role of the individual or organization (See Section 4.2.2). It was initially proposed that perhaps three categories (builders, evaluators, and, approvers) would be sufficient. It was also suggested that possibly the latter two could be combined into one category, leaving just two high-level categories for consideration. No conclusion was reached; further efforts in examining this proposed concept would be necessary, especially if a formal credentials process were implemented.

4.3.3 Formal versus Informal Implementations

Many of the issues discussed in the later half of the session focused on issues that can be grouped under the major heading of whether the issue of pedigree versus credentials should be formalized within the IT security community. This concept is clearly not new nor is it limited to security. It is something we have in everyday life, both formally (e.g., certified public accountants, doctors, lawyers) and informally (e.g., make of car you chose, the dealer you bought it from), or even somewhere in between (e.g., doctor referral service, consumer reports).

Whether formalized or not, this concept is a channel of information that is currently used, albeit on a more informal basis. We listen to those we have come to trust; many do buy products/systems based on who was the developer or integrator, for example.

Many correlations can be drawn to other professional occupations. However, differences do exist. One example cited during the discussion was the construction of a bridge. Clearly, the engineers have received formal training and are licensed. However, strict, measurable requirements for the actual construction also exist. In the area of IT security, similar requirements are not so easily identified.

Formalizing the process of obtaining credentials of an IT security professional raises numerous corresponding issues. Is it worth the effort for an individual/organization to obtain formal credentials if they are not able to make a tradeoff for something else; (i.e., not having to produce as much evidence as the individual/organization who does not have formal credentials)? Liability concerns are an equally important issue. Much insight could be gained in identifying other significant issues from further comparisons to other occupations that currently have a formal process for obtaining credentials as well as those that do not.

SECTION 5

SECURITY ARCHITECTURE AND APPLICATIONS

Judy Froscher, Moderator
Jay Kahn, Recorder

"Our paradigm for managing information security must shift from developing security for each individual application, system, and network to developing security for subscribers within the worldwide utility."

Joint Security Commission Report, 1 March 1994

5.1 INTRODUCTION

The twofold purpose of this session was to identify the opportunities for managing information security for "subscribers within the worldwide utility" and to discuss how architecture contributes to system assurance. The starting point for this session was the report of the JSC, *Redefining Security* (JSC, 1994). While not endorsing or even considering every finding in this report, the session participants agreed that the report provided challenging objectives and should be used as a source of provocative ideas.

Chapter 8 of the JSC report, "Information Systems Security," clearly identifies networking and distributed systems as essential to tomorrow's architecture. Further, the report notes that as a nation, we can no longer afford to develop unique solutions to what appear to be standard problems.

The moderator identified the following three security challenges from the JSC report, which were used as long-term national goals during the workshop discussion:

- Encourage distributed architectures.
- Discourage stovepipe[2] solutions.

---

[2] A stovepipe system has a low degree of horizontal integration with other systems in an enterprise, and a high degree of vertical integration. That is, it does not share or communicate well or at all with other platforms' resources at the same levels in their respective architectures. By vertical integration, a system may have its own dedicated displays, computer hardware and software, communications lines, phone system, sensors, etc. Stovepipe systems have the connotation of potentially never being used again. The term has a pejorative connotation of a system potentially never being used again. It is

- Discourage multilevel secure (MLS) stovepipe systems.

With this introduction, the following five questions were used as discussion points for this session:

- Is understanding and implementing security in distributed systems less achievable or just more difficult? How does distribution increase complexity?
- Are we focused too much on operating system (OS) security? Do we need trusted applications?
- How can security architectures ease assurance arguments?
- Can some security architectures allow integration of new technologies, support more secure use of legacy systems, and promote assurance?
- Can different security architectures allow users more effective access to their data?

While these goals and questions framed the security architecture and applications discussion, most session participants had not considered how security in the large could be achieved. Hence, much of the discussion centered on understanding requirements and how emerging technologies could support security in a worldwide, distributed computing environment. Most participants were much more at ease in postulating security solutions for dedicated applications or for small confederations of simple, homogeneous systems, to which more conventional MLS approaches are applicable. Security architectures that could promote the management and protection of information in a worldwide information infrastructure proved too much of a paradigm shift for this forum.

---

often associated with a program-specific one-time solution that fit the exact system or situation at hand, and is not interoperable with other programs.

The primary advantage of stovepipe systems is performance. They are dedicated to one use and offer maximum availability. There are also advantages for developers. By having all their own vertical components, they may be faster to build as they are independent of common architecture definitions and therefore avoid politics. The disadvantages are generally related to maintenance and extensibility, and they can be more expensive than general purpose systems.

The group reached a consensus definition of *security architecture*. A security architecture is the structure of protection mechanisms that allow the enforcement of a security policy. Security architectures are not unique. The same security policy can be enforced using different security architectures. That is we can rely on different compositions of mechanisms to enforce the same policy. Our goal is to define security architectures that promote the enforcement of a security policy and make the assurance argument as straightforward and simple as possible.

Security architectures can ease the assurance argument by explicitly identifying the role of different parts of a system in enforcing a security policy, identifying dependencies among different parts of the protection mechanisms, and allowing focused assurance strategies for each security-relevant part of the system. Security architectures allow us to compose systems to enforce a policy that governs the security behavior of the combined systems.

5.2 NEW TECHNOLOGIES IN DISTRIBUTED COMPUTING

Two current technologies that could support distributed architectures were highlighted. The first of these technologies involves Object Oriented (OO) development paradigms, as manifested in OO designs, OO data management systems, and OO programming languages. There is a subtle difference between OO design and development methodologies and actual OO systems. The blurring of this distinction is quite confusing. While it is unclear at this time what are the consequences of embedding security attributes within objects, the highly dynamic nature of linking and re-linking of objects is a cause for security concern. This dynamic linking and re-linking of objects is quite different from traditional systems in which security architectures and policies can be regarded as static.

What supporting security mechanisms will be required to interface with the object's embedded security parameters is an unresolved question. The entire success of OO technology itself is open to question. Other similarly promising technologies have emerged in the past without having significant long-term effects on automatic data processing (ADP) development. The security implications of OO technology are currently under investigation in several R&D efforts, but the efforts are too immature to predict any results.

The second technology addresses client-server systems. Client-server technology has already captured a significant portion of the ADP market. This technology has the commercial advantage of being cost-effective in replacing aging mainframe systems.

From a security perspective, client-server technology will introduce difficulty as it will require parts of the security policy to be enforced in different, sometimes heterogeneous, hosts.

However, a consistent access control policy can be used to manage data access servers, while complex applications with application-specific and user-specific interfaces can run on powerful client processors. Perhaps if these applications transmit information at only one security level, clients can operate without trust from a confidentiality standpoint. However, accommodating policies of assured delivery, non-repudiation, separation of roles, or availability will increase difficulty. The employment of widely used, commercially available application software can reduce the risk of integrity vulnerabilities. The seemingly unmanageable interactions among clients and servers can be controlled through transaction management, which can guarantee that many of the integrity concerns that have been discussed in this session do not result in chaos and unpredictability.

The goal of this workshop session was not to solve all of the security problems inherent in either of these two emerging technologies, but rather to recognize that any evolving security development will have to function in either or both of these technological environments.

5.3 IS UNDERSTANDING AND IMPLEMENTING SECURITY IN LARGER DISTRIBUTED SYSTEMS LESS ACHIEVABLE?

5.3.1 Policy Issues

We observed an expanding need to have policies other than confidentiality. These include polices for integrity, availability, and anonymity. We need to have some attributes that characterize both users and data, and that can be used to make decisions about accesses between them. The semantics of these attributes and the attendant accesses depend on specific security policy objectives.

As our computing environment evolves toward a distributed one, it appears that it will become more difficult to assume single, uniform policy coverage. Distributed systems will require multiple security policies and enforcement in multiple domains. These policies will have to be enforced within the system's primary domain, and be applicable, or at least not violated, while in other domains. This multidomain, multipolicy environment will lead to the development of metapolicies that tell us how to make policies. The concept of metapolicies introduces complexities that are not scalable from the current environment. This concept also provides initial evidence that security for distributed systems will be a much more difficult problem than found in a traditional single system TCSEC environment.

Once metapolicies are introduced, the follow-up question must be "Where are decisions made?" We identified three possible answers:

- At the developer's facility
- By the system manager when the system is loaded and configured
- Dynamically as necessary

If decisions are made dynamically, it appears that artificial intelligence engines may be required to provide automated capabilities exceeding the level of sophistication available today. However, if decisions can be pre-determined, security solutions become possible. Again, the complexity inherent in implementing a security policy that can be tailored at the site also increases the difficulty of providing a distributed security implementation.

It has been recognized that there are problems remaining with creating rather simple security policies. Policy-makers desire clear, concise rules. If rules can be defined that are appropriate in every contingency, then the policy-maker has succeeded. However, sometimes policy is only a mechanism in disguise. Policies are rarely complete or concise, nor are they always applicable. When we create security policies, we document and circulate them believing that the policy statements reflect a pragmatic approach for doing business. As long as the users are able to perform their jobs without feeling hindered, frustrated, at risk, or foolish, they nearly always adhere to the policy. However, if users feel that policy interferes with the mission, generally it is the policy that is discarded rather than the mission. Examples of user resistance include the work-to-rule industrial action, the unauthorized sharing of restricted data, or the undocumented disregard for policy.

The conversation described above highlights the issue that we use an imprecise medium, the English language, to document policy statements. The policy that we write reflects the way we want things to operate, and it may be self-contradictory, incomplete, or ambiguous. Further, the policy description defines expectations of how we think we would like the policy to be. A systemic review, such as one conducted as part of a reengineering review, might lead to a different statement of the security policy.

Security policy, like most policies, is often expounded by upper management. These policy statements are interpreted by middle management as rules to be implemented, while conforming to and often preserving the existing middle-management corporate view. This understanding of how policy is made and implemented raises the following question: Is there really a difference between high-level statements about desirable goals and security policy rules? There are concerns in that these differences can be important, that we are striving for nebulous goals instead of crisp, clear policies, and that there is no ideal methodology for separating the two concerns. Ultimately, the issue is one of control: Who, then, is right, the management or the system's designers?

To implement high-level policy for an actual system, the system designer must first decide which enforcement mechanisms must be automated and which will be enforced procedurally or by trusting personnel. to behave ethically. These decisions must be consistent with policy enforcement throughout a distributed confederation of autonomous, heterogeneous systems. The challenge is to make design and policy decisions that promote lower risk solutions.

5.3.2 Access Issues

While access determinations eventually evolve into binary decisions, to make these determinations when multiple policies are in effect, it may be necessary to use new kinds of mechanisms. A candidate mechanism is fuzzy logic[3] . It was noted that fuzzy logic is widely used today in real situations, but the fuzzy logic processor is a human being. Implicit in this discussion is the realization that users need complex security access rules involving factors such as day of the week, time of day, the user's physical location, and the user's role.

5.3.3 Scalability Issues

The issue of increasing complexity with increased scale is subjective. However, it appears that as we look at scaling up from a centralized system to a distributed system or system of systems, it may be easier to implement some security mechanisms because physical separation can provide some enforcement. However, this enforcement is not sufficient for a complete security policy. Distribution appears to add risk, which increases complexity and is not scalable. In general, increased scale brings increased risk.

It was noted that TCB techniques do not seem to scale upward. Meeting applications assurance requirements become too expensive, so we need new assurance techniques that can handle aggregates of distributed data. To find these new mechanisms, we must consider whether access mediation is imposed at the correct level in an architecture. In scaling up for size and complexity, we may need to place protection mechanisms at different levels and apply additional protection mechanisms once a coarse access mediation decision has been taken.

Another possible solution is to increase the security perimeter. This topic is discussed as part of the question addressed in Section 5.4. However, we are beginning to recognize that we cannot afford distributed TCSEC solutions, so we must begin to look at other approaches.

---

[3] Readers unfamiliar with fuzzy logic are referred to Lotfi A. Zadeh; Fuzzy Logic, Neural Networks, and Soft Computing; Communications of the ACM; March 1, 1994, v 37, n 3, page 77.

5.4 ARE WE FOCUSED TOO MUCH ON OPERATING SYSTEM SECURITY? DO WE NEED TRUSTED APPLICATIONS?

5.4.1 Focus on the Operating System

Distributed systems have fundamental problems with some of the features that are required by the TCSEC, for example, *trusted path*. The communications protocols in use today do not support these kinds of features. On the other hand, no product TCB is free of covert channels, although some products now coming on the market claim this distinction. Covert channels have been found in all products having a TCB, even high-assurance products. Covert channels introduce a vulnerability that can be exploited to violate the security policy. Whether to guard against the exploitation of this vulnerability for a given environment is a risk management decision. However, when we move to a distributed computing environment, covert channel vulnerability offers a much richer opportunity with a lower probability of detection for gaining unauthorized access to sensitive information. Access to real threat information can perhaps provide pragmatic guidance for the real risk that covert channel vulnerability poses for the compromise of sensitive information.

Work is being done in maintaining security across multiple domains. European Computer Manufacturers Association (ECMA) Standard ECMA-138 (ECMA, 1989), which addresses the propagation of security policies across domains, was suggested as a useful reference document. With these types of problems in mind, it became obvious that we need to find ways to make security architectures work for and not against us.

Distribution adds both difficulty and complexity to the security problem. As noted previously, we must be able to enforce security policies such as integrity, availability, and safety as well as the traditional policies of confidentiality. Using the security infrastructure commercially available today as a consequence of National Security Agency (NSA) support for evaluated security products, it has not yet become practical to construct a modular architecture from the building blocks provided by these evaluated security products. An architectural methodology would permit the development and combination of these logical building blocks to support all of these security policies, as well as the support of distributed functionality.

New security paradigms must provide services that support enforcement of these policies. These services are the logical building blocks just mentioned. Examples of these services include data abstraction, layered security services, and refinement support. These services were applied to the THETA system (McEnerney, et al., 1990).

Data abstraction is an important software engineering technique for developing quality modular software. It allows assurance claims to be made that trusted software has a limited

impact and hence does not violate the system security policy. It can be used to demonstrate domain isolation and show if new covert channels have been introduced. Data abstraction can also be used to enforce fine-grained access control policies with richer access semantics.

Making security services available to trusted applications when they are needed can be far more effective than providing the service as part of a centralized TCB. These security services can be designed, implemented, evaluated, and stored in a repository. A trusted application uses only those services needed for the application. Examples of these layered security services include mandatory access control (MAC) checks, audit, trusted path, and scheduling.

Refinement support to some extent depends on data abstraction and the availability of layered security services. A trusted application may require a finer grained access control mechanism that can be enforced using data abstraction and a MAC check. The assurance argument is facilitated by demonstrating that the application code satisfies the conventions of the layered security services it uses, and that the code enforces the application-specific security policy.

5.4.2 Trust in Applications

We can gain assurance about security policies other than confidentiality if we trust some applications. By extending trust in selected applications, we gain enough assurance to permit the use of lower assurance systems in environments where other guidance, such as the Yellow Books, may establish a requirement for higher assurance systems. We gain this assurance in the following way. Some security enforcement mechanisms can be implemented in trusted applications. If we place a few restraints on those applications, we can have assurance that this trust is warranted.

The primary restriction is that the application must be trusted not to behave maliciously. We must have assurance that trusted applications do not exploit covert channels, subvert accountability mechanisms, or corrupt information that is processed by the application. These trusted applications become extensions of the TCB. In turn, the TCB must ensure that these applications are not bypassed, are tamperproof, and must provide a trusted path from the application to the user.

There are several proposed methods for gaining assurance that an application is not malicious. One method advocated by John McDermid is a trilateral plan for using professional people, code inspection and analysis, and testing (McDermid, 1991). Others advocate gaining assurance through the pedigree of the investigator, his/her organizational allegiance, or the reputation of the tools used to examine the software. Pedigree was addressed as a separate workshop topic (see Section 4). By gaining assurance in this way, even medium-assurance TCBs with trusted applications can be used in place of high-assurance systems.

Although some distributed system examples were presented, most of the discussion focused on trusted applications running on special-purpose OSs and the sometimes contradictory nature of policy objectives from different security domains. Issues of securely sharing information among autonomous systems, the atomicity of user-generated transactions, the secure handling of congestion among trusted systems, the proliferation of covert channels, and secure recovery in a distributed subscriber information infrastructure environment were not discussed at length, but remain important topics for further investigation.

5.5 HOW CAN SECURITY ARCHITECTURES EASE ASSURANCE ARGUMENTS?

While this question was not explicitly discussed, it was recognized that part of the response to the question addressed in Section 5.4 included this subject. Additionally, if we can separate the assurance problem into smaller, more manageable components, we can reason about these components with more confidence. Defining rules for composing the assurance for these components into assurance for the larger whole is an area of ongoing research. System- or infrastructure-level refinement techniques for allocating requirements as well as assurance objectives also requires further investigation.

Some efforts in DOD address assurance by certification and accreditation of the information infrastructure. The common approach is to define a flexible security architecture that allows individual systems to connect in well-defined ways and constrains the risk and the role that a given system plays within the infrastructure. By defining connection rules, a system's security posture can be protected against perturbations elsewhere in the infrastructure. Only when a system's own security configuration changes does the security posture of the system itself need to be reassessed. This approach makes security management tractable from the system perspective. However, deriving a security architecture in the large, and reasoning about its assurance and protection effectiveness remains a problem for future investigation.

5.6 CAN SOME SECURITY ARCHITECTURES ALLOW INTEGRATION OF NEW TECHNOLOGIES, SUPPORT MORE SECURE USE OF LEGACY SYSTEMS, AND PROMOTE ASSURANCE?

5.6.1 New Technologies and Legacy Systems

In light of the current migration away from legacy mainframe systems toward distributed computing environments, legacy systems require some special consideration. This migration will occur slowly because resources are not readily available.

The Naval Research Laboratory (NRL) has used physical separation, distribution, and replication with a high-assurance product in a prototype system called the Secure Information Through Replicated Architecture (SINTRA) project. A trusted front end mediates access between users logged in at different security levels to databases at their respective login level. Each database contains information appropriate to the login level as well as copies of all lower level information. Users can retrieve information with high assurance and little security overhead. However, when a user updates a low-level database, the update must be securely and consistently propagated to all higher level backend databases.

Most of the research effort for the SINTRA project has been focused on the development of a correct replica-control algorithm for the consistent replication of data, while providing secure, concurrent access to users operating at different levels. Several such algorithms were developed as part of this effort. The algorithm currently implemented is untrusted, and it does not require changes to the commercial database management system running on back end processors. A proof-of-concept prototype has been implemented and demonstrated that high assurance and good performance are not mutually exclusive.

This pragmatic approach to security offers strong security protection, high assurance, good performance, full relational database capabilities, and the ability to incorporate new American National Standards Institute (ANSI) Structured Query Language (SQL) compliant technology. If we can define what a transaction does in a legacy system and what an update implies, the replicated architecture approach can support high-assurance distribution of legacy systems. Likewise, this approach can be used to provide an MLS capability for new technologies such as object-oriented databases, extended relational databases, and expert systems.

In a similar spirit, other pragmatic security mechanisms can be used. In addition to enforcing a confidentiality policy, some form of cryptographic checksum can be used to ensure that information has not been inadvertently or maliciously changed. Intrusion-tolerant mechanisms can be used to thwart the efforts of an intruder and to increase the work factor needed for gaining access to sensitive information. This is an attractive approach for protecting databases against unauthorized access.

5.6.2 Administration of Distributed Security

The discussion addressed the administration of distributed systems. The example was the AEGIS system. In this distributed system, processes can be started on a given hardware platform but can be moved to other platforms to maintain system availability. In this example system, the security policy enforcement mechanisms must also move from platform to

platform in a transparent but highly dynamic manner. This physical separation and mobile security enforcement introduce new security problems.

It is worth noting that on a different level, the management of security across distributed hardware becomes significantly more complex than for a traditional single-processor system. Besides complex issues of synchronizing user names and passwords, hardware-naming conventions, and coordinating audit on-and-off switches, there are major problems with the collection and analysis of audit information. Today's commercial off-the-shelf (COTS) tools meet very few of these security needs.

One aspect of distributed system security that has not received much study is that of multiple identities. Each of us has several identities. These identities might include:

- Joe Sixpack, a commercial network customer
- Joe, a government employee using the Internet at work
- Joe, the office computer-system's database administrator
- Joe, the Parent Teachers Association volunteer at the local school tutoring students in computer skills

For each of these roles, Joe might have a different user name and password. The capability for Joe to access his own files, regardless of his active Internet identity, is problematic. The most straightforward solution is for Joe's access to be determined based on the role for which he is currently authorized. A metapolicy may require that only one role at a time must be active, although this requirement could be very difficult to enforce. This requirement would prevent the following violation of the least privilege policy: in an extreme case, Joe could attempt to satisfy a two-man rule and being both of the required people by simultaneously using two of his user names. Writing a security policy that covers this situation is extremely difficult. Implementing it could be even more difficult. The challenge is to create such a security policy without embedding a solution into the architecture or building a solution into the criteria. These problems illustrate the complexity and conflicts that can arise in the enforcement of multiple policies.

5.7 CAN DIFFERENT SECURITY ARCHITECTURES ALLOW USERS MORE EFFECTIVE ACCESS TO THEIR DATA?

Due to time constraints, this question was not addressed.

5.8 CONCLUSIONS

It was agreed that there is a need to go beyond the TCSEC because this document is not based on the right questions for the new computing world of today. Solutions must be based on understanding the damage that could result from compromise, including confidentiality, availability, integrity, and authenticity compromises; the threats that could cause compromise; and the countermeasures that are effective in protecting data against these threats.

We previously asked "How do we identify the new rules and criteria? Do we allow customer agencies to set their own requirements?" Protection profiles can be used to describe the needs of a particular industry or industry segment. However, there is a pressing need for at least a few of these industry segments to create protection profiles. The first published profiles will greatly contribute to the understanding of user security needs in a distributed environment and to preliminary determinations of which functions need automation.

Recalling our original challenges as stated in the JSC Report, it is essential that the information security (INFOSEC) community must begin to study these problems from the subscriber's point of view. The subscriber will access information from a vast collection of heterogeneous data sources. As our dependence on increasingly greater amounts and varieties of information grows, our ability to manage, manipulate, and protect information becomes more critical. We must be able to ensure that changes made to related information result in consistent information and that consistency is preserved even when some components fail. These observations lead us to conclude that the effects of a subscriber's input request, or transaction, either become permanent within the distributed computing environment or that no effects of the transaction persist. Hence, the transaction becomes the control abstraction for a distributed computing environment. This concept enables us to build upon the INFOSEC infrastructure and discipline that have been developed over the past two decades.

Security decisions must be made that ease the transaction management problem for distributed, heterogeneous information sources. If a subscriber's access must be mediated over some collection of MLS stovepipe application systems, we must impose MLS and application-specific constraints on an already difficult transaction management problem. The challenge for the INFOSEC community is to initiate investigations that lead to a better appreciation of distributed computing environment problems and to provide guidance that makes solutions in the large more possible.

We must appreciate that the transition to a subscriber-information infrastructure computing environment presents as many opportunities as challenges for INFOSEC solutions. Stovepipe

MLS solutions have focused on constraints and restrictions. Application of security measures in the large can provide authorized users with secure, reliable access to all the information they need to do their jobs.

SECTION 6

PROCESS

Joel Sachs, Moderator
Caralyn Wichers, Recorder

6.1 INTRODUCTION

Background information is presented to both aid in understanding and to provide some context. This information is based on the moderator's presentation. Process was defined as the set of practices, methods, and transformations that integrate managers and engineers in using technology to attain an end-result. The fundamental challenge to organizations today is to develop quality results, both reliably and predictably. Key leverage points are people, technology, and process. Unfortunately, the role of process has been given minimal attention to date.

The session discussed the fact that many interrelated processes exist today; these may be formally stated or conducted in default. Major process areas include: acquisition, integration, development, product evaluation, system certification, and system accreditation. The names for these process areas may differ among the military, civil government, and private sectors, but the basic notions are universally applicable. DOD-oriented terms are used throughout this section.

Specialty processes relate to specific disciplines such as systems engineering, software engineering, hardware engineering, test engineering, security engineering and its associated process, operating systems, and accepting risks. Some disciplines cut across the major process and as a result, one can view either a specialty process within a major process or one that branches across them. While a single integrated process sounds attractive, it most likely would be too complex, too high-level, and ineffective. Managing multiple processes in concert is the challenge of concurrent engineering management today.

Processes can be thought of as branching across or constrained within life-cycle phases. Regardless of development approach, system/product projects go through concept definition, design, implementation, and testing in some form or another. Today several development approaches are in use or under consideration. These include evolutionary acquisition, incremental build, prototyping, and "classic" waterfall. Equally important is considering whether a system/product is built from scratch, re-engineered, integrated from 100% COTS, or integrated with developed applications.

The term *process* can connote many different things, including: process definition, process description, process (activity) prescription, process practice, process enactment, process improvement, and process measurements. Directly related to the process measurement is capability assessment and capability models to perform such assessments.

6.2 MAIN CONCEPTS DISCUSSED

6.2.1 Why Focus on Process

There are a number of reasons to emphasize process today. We observe that systems and products apparently will continue to increase in size and complexity. In addition, they will transition through various versions and releases, most likely more rapidly. Their use, environment, and re-use will evolve. Single entity systems will become more a part of an infrastructure, which will become more a part of a National Information Infrastructure (NII) or Defense Information Infrastructure, which become more a part of a Global Information Infrastructure. Such demands and timeliness will necessitate more reliance on the actual engineering of the products and systems that comprise these and, in particular, a need for reliable security engineering to be conducted constantly. Process improvement and assurance are critical to such a need and perhaps the only feasible solution.

Focus on process focus introduces the possibility for scalability, knowledge evolution, and improvement. Such focus will help predict and guarantee predictable outcomes, trends, and characteristics. In addition, it will concentrate investments to enhance and perform quality and effective security engineering.

6.2.2 Reliance on Process

One part of the discussion focused on the relationship between the process and the amount of assurance provided by a system/product. An interesting point was realized in asking the question: To which process are we referring? Some said the process for building something, whereas others mentioned the process for assessing something. The process for operating the system/product may also support the ability to determine assurance. These may be fundamentally different assurances or may be different ingredients to a single notion of assurance.

Before one can agree on a uniform process for determining an appropriate amount of desired assurance and the ability to measure assurance, a common dictionary for customers to use is really necessary so everyone can communicate effectively and consistently. We need to specify what is desired, not how it is implemented.

Another part of the discussion focused the benefits in using a uniform process. Uniformity, both within an organization and across organizations, provides some increased confidence that a particular process is employed properly. Moreover, uniformity in process may lead to improved evidence.

The following additional points/questions were made regarding uniform process and reliance on process:

- Running a set of conformance tests do not adequately assure a system/product as testing cannot be complete or comprehensive, hence process must address more than just testing.
- There is a difference between the individuals or organization qualifications to certify/evaluate versus responsibility for certifying/evaluating.
- While a process is being followed uniformly, there will be pressures to deviate from it, for example from the accreditor, PMO, integration PM, etc., findings will need to account for adjustments in the process.
- To what extent should a security knowledgeable individual be involved in the process in order to be able to address the security issues?
- How can we ensure that the evidence will provide assurance when needed during the process?
- How can tools help?

6.2.3 Measurement and Assurance of Process

Documenting a process is an insufficient way to conclude that the process is followed. It is not only critical to know that a process is being followed, but also to know how well is it followed and how sophisticated is it. Hence, there is a need to measure and assess the quality and degree of process adherence within an organization.

Three major security-related processes and their execution:

- The *engineering process* defines the security requirements, develops a security architecture and design, conducts security testing, and collects and presents evidence

about the system/product (usually executed by the engineering organization, typically a system integrator/developer or product (vendor engineering group).

- The *assessment process* examines the evidence on the system/product and the activities of the engineering organization (usually executed by a system certification or product evaluation organization).

- The *accreditation process* accepts and approves the operational risks associated with the use of a system and its inherent weaknesses (usually executed by the accreditor).

One process model discussed was the Security Engineering Capability Maturity Model (SE CMM). This model focuses on the development engineering activities which are security specific and their interface to other areas, e.g., evaluation, certification, acquisition, and quality assurance. It views security engineering as an engineering discipline conducted concurrently with other disciplines, e.g., systems engineering, software engineering, hardware engineering, and test engineering. The overall result is confidence in the organization's process and their adherence to it.

The following additional points/questions were made regarding measurement and assurance of the process:

- What minimal things do you need to do to make sure the process is applied and followed?

- With what initial set of things does a security engineering process need to start, e.g., threats, risks, mainstream vulnerabilities, etc.?

- Process, technology and people will change.

- How does a risk manager/taker rely on process assurance?

- How should other key processes be addressed, e.g., product evaluation, system certification, system accreditation, system acquisition?

6.2.4 Understanding Assurance and Its Ingredients

An assurance taxonomy was introduced to aid in discussing and understanding assurance. The elements of the taxonomy are: the target, the method, and the benefit of the assurance. The target needs to be considered both in terms of the explicit (immediate) target and the

(ultimate) end target. Three aspects are important regarding method, namely the production method, the assessment method, and the results representation. Benefit needs to be thought of relative to direct benefit and indirect benefit.

Various aspects of assurance were discussed throughout the session. These included seeing assurance in terms of attributes, degrees, and ingredients. The attributes discussed covered correctness (strength of mechanism and quality of their implementation), effectiveness (how well mechanisms do their job), workmanship (quality of development and overall quality of end result), and usability. The possibility for other attributes was acknowledged.

Concerning degree of assurance, it is necessary to recognize that assurance is a continuum. Potential ingredients to establishing degree of assurance could include functionality and engineering, quality control, and assessor process and evidence.

Many of the elements of the taxonomy can be seen as independent of each other. However, it was clear from the discussions that the real interrelationships of them is unknown. Moreover, how to establish and express the type and degree of assurance needed is not really available today. Current practices are weak. The degree of structure required to address them was not clear. All of this is further complicated since risk management is interwoven with politics. A suggestion was made to examine the taxonomy in terms of the true added value of the evidence and the assurance. These may imply that multiple types (representative forms) of assurance, ratings, and evidences are needed to satisfy multiple types of consumers.

The following additional points/questions were made regarding understanding assurance and its ingredients:

- Depending on the type of the assurance requirements needed, the attributes of the assurance for the process may be quite different.

- The user of information technology (IT) security should decide what the needed evidence is to make the product considered enough of assurance.

- Levels of evidence are different and need to be considered, i.e., incorporated into an accepted model. Examples include reputation, warranty, third-party evaluations, industry or consortium branding, and wide-spread use.

- Effectiveness, correctness, and risks are different for systems than products, particularly what they mean and how important they are.

- How should subsequent discovered bugs and problems be handled and how should they affect one's view of previously established assurance or assurance ratings?

- Assurance can be seen as based on the composition of a variety of evidences.

6.2.5 Relating Process to Assurance

Clearly a relationship between process, evidence, and assurance exists. Today assurance is predominantly determined by the product evaluator's process or the system certifier's process for products and systems, respectively. Also today the developer's process (whether a product or integrator/developer engineering group) contributes directly to the creation of evidence to be used in an assurance determination. Therefore, higher quality and improvements to these processes can result in cheaper, faster, better, more predictable assurance.

The following additional points/questions were made regarding relating process to assurance:

- With a high quality process(es), the amount of system/product specific evidence can be greatly reduced.

- Process-produced evidence needs to include effectiveness.

- Is the assurance statement related to the work factor, i.e., degree of effort? If the work factor is large then is assurance improved?

- There is a need to determine how the process can generate the evidence required by the customer then determine.

- Variations must be allowed in the specific practices of a process (i.e., instantiation and implementation).

- In reality, there will be cases where a product is developed without following a defined process and these cases cannot be ignored, i.e., will need to be accounted for.

6.2.6 Process Assurance as a Basis of System/Product Assurance

The discussion at the session centered around the extent that assurances about process could contribute to assurance about the system/product. As stated above, clearly process contributes to evidence upon which classic assurance is determined. It should be equally clear

that assurance about a relevant process, e.g., the developer's security engineering process, can also contribute to the evidence. For example, such assurance would indicate a degree of confidence in the evidence contributions produced by the process.

One could argue that today's system and product assurance are based on assurance of the product evaluator's and system certifier's processes, respectively. The assurance on each process is a forgone conclusion offered by its practitioner. Moreover, these process assurances are usually quite accepted by the consumers, accreditors, developers, and users.

The most challenging question is whether one could rely on developer's security engineering process assurance as the sole or predominant determinant of system/product assurance. Almost all of the attendees felt that relying on a good process alone to achieve an amount of assurance isn't adequate. For instance, testing would still be needed to ensure that process is followed before, during, and after the system/product is made. Total dependence on the process is not enough, because there may be errors and such dependence appears naive. The SE CMM was viewed as a promising method (and criteria) for establishing security engineering process assurance.

The following additional points/questions were made regarding relating process to assurance:

- Process assurance may be adequate to be the sole determinant of system/product assurance.
- Comparisons and equivalencies of various blends of developer process assurance and system/product evidence to establish system/product evidence are needed.
- The Software Engineering Institute (SEI) CMM for Software, while providing assurance on software engineering management, does not address security.
- ISO 9000 fundamentally requires the developer to have a documented process; it does not give any details or criteria.

6.2.7 Process Improvement

System security engineering organizations currently have differences in the maturity of the processes that they follow. Immature organizations tend to use processes which typically do not provide a great deal of visibility into the progress and quality of the system/product being built. These processes are indicative of unpredictable performance and lead to excessive maintenance costs.

The SE CMM identifies key process areas and provides the ability to determine the strengths and weaknesses of the process. Existing relationships between the key process areas are utilized. It provides an approach to improve the process for building a system/product. This is done by identifying incremental improvements and focusing investments in training, tools, and process development.

The group basically agreed that if the process to build the system/product can be improved, then the product will be improved, and therefore reducing the amount of required system/product specific evidence. To improve the process of gaining assurance in a system/product we need to identify the processes that mitigate the risk and relate them to the process that develops the system/product. After you define those processes you need to differentiate the kinds of assurances related.

The following additional points/questions were made regarding process improvement:

- How much evidence will be needed for illustrating that you follow a particular process?
- How much additional evidence above and beyond following a particular process is required to indicate a sufficient amount of assurance?
- What is the cost of improving the process?
- What is the cost associated with evaluating the additional evidence?
- What are the relationships between the risk and methods used to counter the risks?
- How can we disassemble threat and the relationship of the process?
- How can tools help?

6.3 CONCLUSIONS AND FUTURE DIRECTIONS

Clearly movement to a process orientation for engineering, systems, and products is a necessity. Integral to such a movement is a commitment to continuous organizational process improvement, where slow incremental organizational process improvement is key. Such notions can apply equally to processes for security engineering (system or product), product evaluation, system certification, system accreditation, acquisition, and system administration.

Improved and uniform process(es) may lead to higher quality and more predictable evidence and, hence, better, faster, cheaper assurance. Process assurance will likewise aid in this direction. Hopefully, process assurance will permit a reliance on process as the predominant determinant of system/product assurance, perhaps even the sole determinant.

Such reliance will need to be investigated, resolved by analysis, trail, or a combination thereof. The various questions and issues merit investigation:

- A better (practical, usable) definition of assurance with more robust taxonomy that addresses assurance attributes, ingredients, and forms that differentiates assurance purpose and degree.

- Understanding of the trade-offs and the translation of them to equivalency classes; these must address trade-offs among security functionality development process, assurance process, and evidence dimensions.

- Development of the SE CMM and its use in security engineering process improvement, assurance, and standardization.

- Addressing other processes in terms of process standardization, improvement, and confidence.

- Addressing process related issues, such as individual skills, process descriptions, and process prescriptions.

SECTION 7

METRICS AND TESTING

Jim Williams, Moderator
Caralyn Wichers, Recorder

7.1 INTRODUCTION

"What is assurance?" Without a working definition of assurance, it is difficult to describe how to measure assurance. An appropriate definition of assurance depends, in turn, on the rationale for assurance.

Participants in this session discussed definitions and purpose assurance, how it can be measured, both qualitatively and quantitatively, and how testing, including automated testing, fits into assurance.

7.2 BACKGROUND

The following relevant findings can be found in *Redefining Security* (JSC, 1994). A balanced mix of information systems security, personnel security, and physical security is needed, but how does one make sure that an appropriate mix is achieved? A complete range of security objectives needs to be addressed, including confidentiality, availability, integrity, and security management, but do assurance techniques designed for confidentiality necessarily carry over to other objectives?

The *Federal Criteria* (National Institute of Standards and Technology (NIST) and NSA, 1992) provides a view of security in terms of a basic problem decomposition into task areas, most of which constitute assurance of some sort (those that do not are italicized):

- *Product functionality*
- *Environment/usage*
- Product cycle (design, development, evaluation, maintenance)
- Vetting of the above requirements
- *Product integration*
- System certification
- *System accreditation*

The *Federal Criteria* itself addressed only the first four items.

7.3 RATIONALE FOR ASSURANCE

The primary reasons for providing assurance include the following:

- Reduce threats to information assets, safety of physical systems, data sources and recipients, and users.

- Reduce losses induced by active threat agents, natural disasters, faulty software, incompatible system components, inadvertent "agents," and expenditures on security protection.

There was some discussion as to whether the goal of achieving acceptable security could be understood in terms of the cost of security, including indirect costs such as inconvenience to users and security breaches.

Reducing concepts to dollars may encourage cynicism, such as determining what a secret is worth and who cares, rather than foster making a meaningful decision. However, to achieve any assurance, one must spend money. Even in building a system without explicit security requirements, one still wants to test and show that the system works. Depending on the level of assurance needed, one may want to spend more to obtain it. In some cases, one may want to spend a lot for assurance because one may care about saving lives, for example. In others, any additional expense on a particular security objective, such as confidentiality, may be wasted on the product's intended customers.

There must be requirements for measuring the cost of security. Without these requirements, there is no empirical basis for measuring the payoffs from using various assurance techniques. Unfortunately, we do not often learn from the past whether or not assurance techniques were successful. Based on unsuccessful efforts to elicit the voluntary production of cost-benefit information, it appears that if there is no requirement to record the information, then the vendor will not record it. We need to capture information, for instance, on how long it really takes to test and how this testing correlates with how much assurance is obtained as a result. We need, for example, to obtain information on security tradeoffs between testing prototypes and testing actual systems. Currently, testers ask how much money is involved and how long testing will take, but not how this testing will correlate with likelihood of meeting user security needs.

7.4 DEFINITION AND PURPOSE OF ASSURANCE

The following primary and secondary definitions were developed based on a managed brainstorming session conducted using a Juran tool.

- *Assurance* is confidence that a system meets the security needs of those whom it was intended to serve.[4] Thus, the purpose of assurance is to mitigate risk that security needs will not be met.

- *Confidence*, in this case, varies from an informal belief, comfort level, or sense of well being to rigorous, statistically valid measures of probability of truth. It is difficult to see how meaningful quantitative measures of confidence can be obtained without resorting to rigorous statistical notions of confidence. For many people, confidence implies substantiating knowledge of how the belief was obtained, by employing of a precise, rigorous analysis of system behavior, having confidence in the people who perform the analysis, supplying substantiating evidence. Confidence depends on individual perspective; there is usually some loss of confidence as assurance information is transferred from producers of this information to its consumers.

- A *system* is made up of automated information products, users, and other interacting entities. This fact leads to distinctions among automated system assurance, product assurance, and personnel assurance.

- *Meeting security needs* presupposes the full and correct identification of security needs. Thus, assurance necessarily includes validating the correctness and completeness of identified security needs. Moreover, the needed degree of assurance depends directly on the expected cost of failing to meet these needs. Cost is not always measurable in monetary terms. Meeting security needs also involves the proper design and implementation of systems. The overall system design process includes formulation of automated system requirements as well as usage and

---

[4] This definition is significantly stronger than that found in the NSA *Glossary of Computer Security Terms* which defines assurance relative to an assumed system security policy that, in some cases, may be irrelevant to the security needs of people whose lives may be affected by the system at hand. The *Federal Criteria's* definition, which is closer to ours, splits assurance into profile assurance and IT product assurance. Profile assurance is equated with technical soundness, which implies appropriateness of the profile's functional and assurance requirements.

environmental constraints. Consequently, assurance involves ensuring that the overall design *effectively* addresses the needs of the users and owners of the system. The overall structure, or design of a system, is something that exists throughout its life cycle. Assurance of effectiveness is needed *continuously* throughout the entire evolution of the system.

- The overall security design must also be *correctly* implemented. Automated components must perform as specified, which implies not doing things that are prohibited by the design's security requirements. Moreover, constraints on the use of the system must be reasonable and must be explained to its users and administrators who, in turn, must have sufficient incentives to obey these constraints. Assurance of correctness must also be provided continuously throughout a system's life cycle.

7.5 KINDS OF ASSURANCE

The above discussion on the purpose and definition of assurance implies directly that security assurance includes the following kinds of assurance:

Policy assurance involves the identification, assessment, and validation of security needs, as well as estimation of the (possibly non-monetary) cost of failing to meet these needs. This form of assurance requires empirical data on security incidents. Like other forms of assurance, this form must take place throughout the system's life cycle because relevant empirical data is generated throughout the system's life cycle and because security needs evolve in parallel with those of the system.

Effectiveness assurance ensures the effectiveness of the overall system and component designs, including the design of environmental/usage constraints, throughout their entire life cycle. Effectiveness, in the ITSEC at least, is the aptitude of the security functions to properly counter the postulated threats. Effectiveness includes suitability and strength of mechanisms. The FC adds adequacy, completeness, binding, and dependency analysis.

Correctness assurance ensures that designs are correctly implemented throughout their entire life cycle. Correct implementation refers to agreement between implementation and specification.

Evaluation assurance provides evidence as to whether policy, effectiveness, and correctness assurance is adequate. Successful evaluation involves expert examination of evidence pertaining to policy appropriateness, design effectiveness, and implementation correctness.

Pragmatically, the above decomposition of the assurance problem requires an appropriate, *balanced* allocation of assurance effort among various kinds of assurance, among system components, including human components, and among various assurance techniques, throughout the system's life cycle. Balance presupposes the identification of all readily available sources of assurance and assurance evidence. The need for allocation of assurance throughout product and system life cycles is a strong motivation for emphasizing *reusable* assurance evidence. Our discussion did not devise a name for this process, but for ease of later reference in this document, we will refer to it as *ensuring balance.*

For all kinds of assurance, including ensuring balance, the amount of assurance actually provided depends heavily on the *expertise* of the people and organizations involved; it is positively, perhaps strongly, correlated with their reputations. Consequently, reputation may be a useful, convenient form of evidence for evaluating all kinds of assurance. For the same reason, willingness to accept responsibility (including legal responsibility) for error is also positively correlated and is thus useful as evidence. In particular, vendor product warranties may constitute evidence of correctly implemented, effective product designs.

All of the various kinds of assurance can fail. Security incidents will happen, and thus assurance will be improved if response to breakage is planned for, especially in regard to the maintenance of policies, systems, and the products from which systems are built.

7.6 ASSURANCE TECHNIQUES

The following subsections discuss in more detail the above five kinds of assurance, along with related assurance techniques. This list of techniques produced during the discussion is clearly incomplete. The apparent absence of techniques for some forms and aspects of assurance may be a significant observation or simply a result of the brevity of our investigation.

7.6.1 Policy Assurance

Policy assurance, as it applies to information security, appears to be an under-explored area. Other similar disciplines, such as industrial safety and physical security, have a stronger tradition of collecting, analyzing, and profiting from incident data, both real-world and experimental data.

7.6.2 Effectiveness and Correctness Assurance

There is a large body of knowledge and assumed wisdom about how to achieve high quality design, implementation, and systems integration. We did not discuss this topic in detail. However, product and automated system development necessarily includes developer

evaluation, so that the evaluation techniques mentioned below apply here as well. Moreover, the overall value of a development technique may include not only intrinsic value to the product or system produced but evidentiary value for evaluation.

Assurance in the design and implementation of environment and usage constraints was touched on only briefly, and less appears to be known than originally thought. Techniques include screening/training users and product integrators as well as redesigning products and systems to make them as idiot proof as possible.

7.6.3 Evaluation Assurance

Available evaluation assurance techniques include, but are not limited to, the following:

- Direct, rigorous analysis of product or system behavior
- Use of formal methods. such as specifications and correctness proofs
- Covert channel analysis (mention of this drew protest on grounds of practicality)
- Penetration testing
- Functional testing (including beta testing)
- Assessment of developer competence and/or methodology
- Assessment of developer reputation
- Assessment of development tools (e.g., for design analysis, configuration management, automated testing)
- Assessment of user experience
- Avoidance of using the first version of a system, which usually involves field testing and evaluation

7.6.4 Ensuring Balance

Assurance techniques often tend to apply to perceived-threat scenarios, focusing on what might go wrong. The assurance approach for a given system or product family needs to be subjected to threat-mitigation or risk-reduction analyses.

Unfortunately, there is a dirth of relevant information here. For example, there is no clear way to authoritatively answer such obvious questions as the following:

- For what environments would money spent on covert channel analysis have been better spent on configuration management?

- Is having an NSA A1 evaluation better than being a COTS product?

7.7 WHERE TESTING FITS IN

Evaluators tend to test what the developers do. If tests performed by the developers were correct and complete, perhaps the evaluators could just check the results.

There are a few areas in which exhaustive testing is already being used, one of which is model checking of hardware. For portions of systems that can be specified in terms of propositional logic, it is feasible to set up binary decision trees to check all paths. This is common practice for testing of hardware; these tests are fully automated.

Security testing traditionally includes penetration testing, which is difficult to automate because it involves long, unusual scenarios. Penetration testing is based on the assumption that the system will have hostile users whose patterns of input are explicitly designed to exploit system vulnerabilities. Traditional testing theory requires that test cases be distributed in the same way as input in actual system operation. Unfortunately, input patterns by hostile users can change when vulnerabilities are discovered—in ways that cannot be predicted during routine testing. There are tools that help with penetration testing, but full automation seems infeasible.

7.7.1 Where Automated Testing Fits In

Fully automated testing involves constructing machine-readable specifications that describe both system behavior and expected user inputs, automated generation of test cases from the specifications, automated execution of test cases by test harnesses, and automated analysis of test results by test servers.

Automated test generation involves additional expense because of the need to write machine-readable specifications. However, automated test generation may be very cost-effective when the following statements are true:

- Economies of scale are realized

- A single design has many different implementations

- Assurance requirements call for
  - Systematic or exhaustive testing
  - Independent validation of vendor test results
  - Machine-readable specifications
- Automated test execution is required for other reasons
- Test-generation tools can develop new test suites quickly for
  - Design revisions
  - New test requirements
  - New implementations

To what extent does automated testing support evaluation? Unconstrained searching for vulnerabilities is a good thing to do but it is not clear if this is feasible via automated testing, if hostile users are assumed to exist. However, automated regression testing can be used to ensure that known security flaws do not reappear after having been removed.

7.8 CONCLUSIONS

Several useful sources of assurance have been underutilized in the past. More cost-effective approaches depend on understanding how these sources can be best used. In this session, we have posed tentative answers to all but the last of the following questions:

- What is the higher goal that security assurance supports?
- How does assurance fit in?
- What is assurance?
- How does (automated) testing fit in?
- How can assurance be achieved and measured?

These answers and the final question need to be discussed by the larger security community, possibly at NIST's International Invitational Workshop on Developmental Assurance, to be held in June 1994.

Finally, we have identified a preliminary taxonomy of assurance elements and techniques. This taxonomy pertains to the difficult questions of how to achieve practical assurance levels and how to measure, either qualitatively or quantitatively, the amount of assurance provided for a product or system. It was the consensus of the discussion group that qualitative measurements of assurance are currently more feasible than quantitative measurements.

SECTION 8

RISK MANAGEMENT

Marshall Abrams, Moderator
Lynne Ambuel, Recorder

8.1 INTRODUCTION

The challenges of risk management are divided into equally demanding and sometimes conflicting requirements concerning data integrity, system integrity, availability, software reliability, safety, and confidentially. Accepting a risk management perspective, we cannot pretend that by implementing certain security measures, we can mitigate the security risk to zero. Viewing risk assessment from the standpoint of assets and threats is necessary but not sufficient.

Participants in this session discussed relevant extracts from *Redefining Security* (JSC, 1994), fundamental questions and terminology, multidimensional complexity, trade-off and balance, scope of IT security, decision-making, system characteristics, contingency plans, and dissemination of information.

8.1.1 Extracts from *Redefining Security*

Like many of the other sessions in this workshop, we found relevant extracts in *Redefining Security* (JSC, 1994):

- Security of information systems and networks is the major security challenge of this decade and possibly the next century.
- The paradigm for managing information security is subscribers within a worldwide utility connected to and dependent upon an infrastructure they neither own nor control.
- In most cases, it is possible to balance risk of loss or damage of disclosure against cost of countermeasures.
- We must use a risk management approach that considers actual threats, inherent vulnerabilities, and availability and cost of countermeasures.

- Risk management requires evaluating the resource impact of proposed changes in security policies and standards.

The members of the working session felt that Figure 1, extracted from *Redefining Security* (JSC, 1994), which shows the risk management process, is an attempt to use the scientific method for a problem that has not been reduced to science.

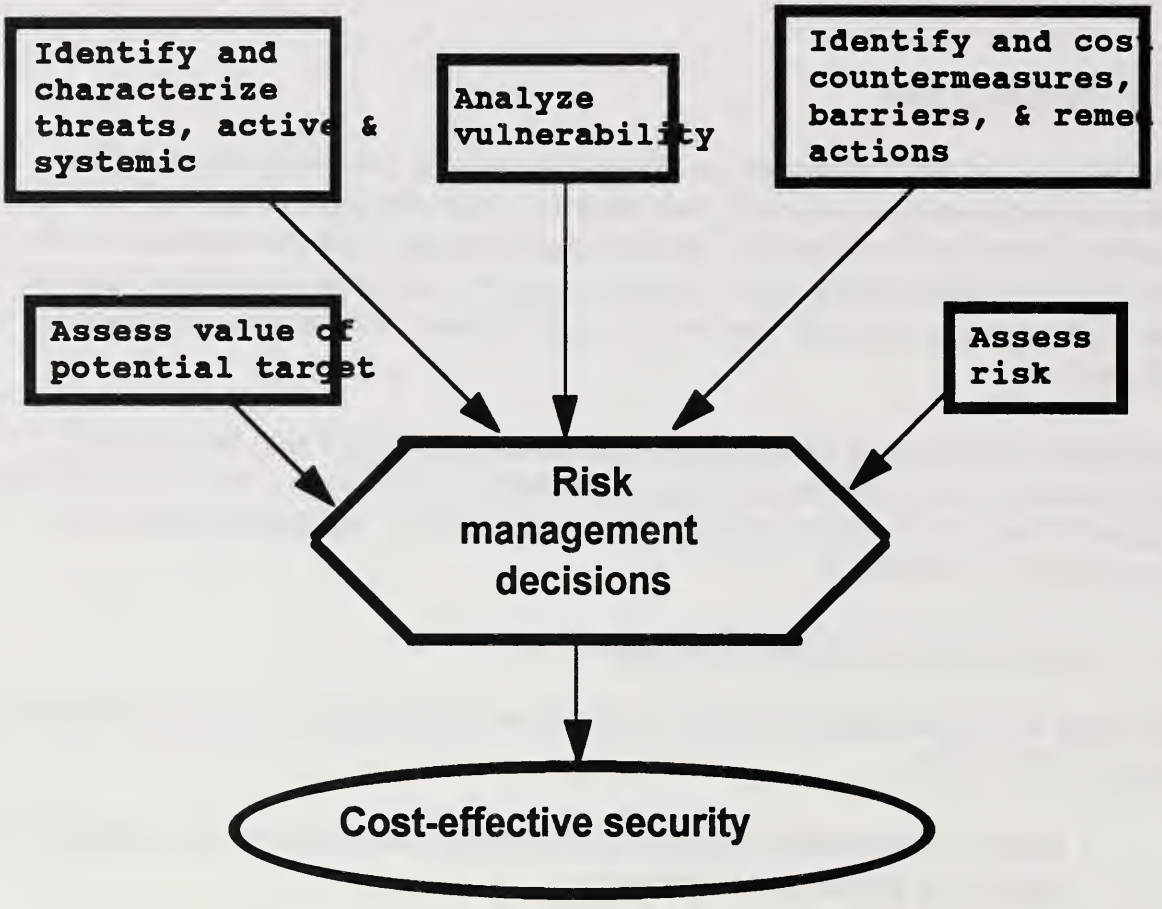

Figure 1. Risk Management Process

8.1.2 Tools Needed for Risk Management

Among the tools needed for risk management are a language to capture requirements and maintain cognizance of requirements throughout a system's life cycle; a risk quantification method that relates to actual requirements, addresses inherent risks, deals with complex implementations, and performs meaningful computations; a methodology to lead designers and evaluators through the full spectrum of risk issues, identifying which concerns are

applicable to a particular system and going into more depth where appropriate; and a listing or rating of risk reducers.

8.2 FUNDAMENTAL AND VERY TOUGH QUESTIONS

Fundamental and very tough questions need to be answered to make any progress on risk management: What are the security requirements? In what ways are we at risk of not meeting those requirements? How much are we willing to spend to mitigate those risks and to what degree? What should be the government's role in helping to protect information held by private citizens and institutions? How can government technology be provided to the private sector for the protection of sensitive unclassified information? Will the private sector accept it?

8.3 TERMINOLOGY

Semantic distinctions and definitions are very much part of the problem faced in capturing and presenting issues in assurance and risk management. Even when terms are defined, it is very difficult to get people to read the definitions and to use the terms as defined. Nevertheless, failure to define terms almost guarantees failure in meaningful interchanges concerning risk management.

This session met this problem half-way. We identified four key terms that must be defined: risk, threat, vulnerability, and susceptibility. The participants were able to communicate based on prior knowledge from working in the field. However, we recognized that there are multiple authoritative definitions which differ among themselves in both subtle and more obvious ways.

8.4 MULTIDIMENSIONAL COMPLEXITY

While the desirability of quantification is recognized, it may not be possible given the state of understanding. Qualitative descriptors may be sufficient and necessary based on whether an assurance factor is quantifiable. In some instances, simple ordering may be achievable and desirable.

One of the consequences of the paradigm switch from risk avoidance to risk management is that it becomes necessary to deal with the unimaginable. Especially in times of decreasing resources, it is necessary to think about what actions should be taken if highly undesirable events occur.

As discussed in more detail below, it is also necessary to consider incomparables in risk trade-offs and management. People may make decisions based on system security, human safety, personal career, or any other factor that they may consider to be important to themselves or their organization. In addition, security risks are only one component of risk management. The risks to development/production schedules, inclusion of competitive, state-of-the-art technology, and sales factors often dwarf the security risks in the decision-making process.

The imperfections of current risk management techniques were acknowledged, but no alternatives were identified. The group generally agreed that, although the techniques and tools for performing risk analysis could be improved, the management of risk will always be an integral part of product and system development. It will be necessary to refine the concepts and practices of risk management as they are applied to information security.

In discussing risk management, it is necessary to distinguish between the general concept of risk management and specific techniques. It is not uncommon for a discussion to be couched in terms of *risk management* when the speakers have specific techniques in mind. Communication can be especially difficult when different techniques are being discussed but have not been made clear.

8.5 TRADEOFF AND BALANCE

There are several kinds of risks to which systems are subjected, including technical, schedule, cost, security, and safety. Satisfaction of all objectives and avoidance of all risks are generally impossible because the objectives or the techniques used to achieve these objectives are often in conflict.

Perspective enhances the ability to make the trade-offs, but the job is never easy. It involves balancing conflicting equities. Sometimes, decisions are made suboptimally because the decision-maker is unaware of all the consequences.

8.6 SCOPE OF IT SECURITY

Traditionally, IT security has focused on products and systems. However, the scope extends beyond these areas in several dimensions. Security can affect the survival and well being of entities, including the individual, the organization, the nation, and the planet.

Integrating products into systems and forming systems of systems are unsolved security problems. It has been recognized for several years that we need standards and procedures for

preserving security attributes and properties through the integration process. Little or no progress has been made on developing guidance or codifying good practice in this area. The security impact of integration remains an art form due, in part, to the subjective nature of risk management.

IT systems exist in an environment. One of the salient characteristics of systems, as defined in the ITSEC and FC, is that a system is used in a specific real environment. The IT system and the environment are real entities that interact. The circumstances and realities of the environment constitute boundary conditions and requirements on the IT system. Non-technical countermeasures are also part of the environment. The opinion has been voiced, but not conclusively established, that non-technical countermeasures are more cost-effective than technical ones. Proving and using this assertion can be both a demonstration of risk management techniques and a tool in the utopian trade-off tool kit.

A significant part of the environment is the information infrastructure. Networks connect resources across agencies, companies, industries, countries, and the world. The laissez faire, cooperative, decentralized federation of the Advanced Research Projects Agency (ARPA) sponsored Internet in the 1980s appears inadequate for the commercialized global Internet evolving in the 1990s. Understanding and responding to changes in this part of the environment are significant problems.

8.7 DECISION MAKING TECHNIQUES

Management has been described as decision-making based on insufficient information, and risk management is decision-making in the face of uncertainty. Risk management can be characterized as technically complex, multivariate, and not fully understood. Existing techniques include sensitivity analysis, system effectiveness analysis, and cost-benefit analysis.

It is not clear whether such techniques are applicable to managing security risk. Data is lacking on whether these techniques have been employed, whether they have proven to be effective, and how acceptable they are to the risk managers.

The models underlying the techniques also need to be reexamined. Are Bayesian, probabilistic, and actuarial statistics applicable? How does the possibility of a human agent attempting to violate system security policy affect the underlying assumptions on which the models are based? Are fuzzy system techniques applicable and acceptable?

8.8 DECISION-MAKERS

Who makes which type(s) of decision(s)? The appropriate person, level, criteria, and method for making a decision is not always clear. Technical personnel, line management, and type management all make decisions that impact risk. Technical personnel may not recognize overriding policy or political concerns, for example.

Accepting responsibility for decisions is complicated by group decision-making techniques, such as mutual and peer decision making. In addition, many organizations have lack of responsibility built into their decision-making processes. In these organizations, decision-makers are on a fixed, short-term tour of duty in which it is essentially guaranteed that a product/system will not be operationally deployed by the time their tour is over. Because of this, the decision-maker may consider career advancement decisions and avoidance of adverse publicity to be major factors is the decision-making process as they will not be directly accountable for the operational product/system. There needs to be more accountability of these decisions, perhaps by placing people into positions so they stay with a project, having responsibility for all stages in the development and deployment of that system.

Qualification of decision-makers is highly variable. Academic training in the referenced techniques is infrequent. Many organizations believe that general managers or general officers can move among disciplines with equal effectiveness.

8.9 BASIS FOR DECISIONS

Managers make decisions based on many considerations. The scope of their influence or knowledge base will shape their decisions both in terms of technical/political risks and localized/global perspective. Career impact is a highly motivating decision basis and may result in a decision that reflects personal accountability and not necessarily technically optimum. The decisions are often made without sufficient information and are therefore somewhat error prone, if not highly subjective in nature.

In addition, there appears to be a tendency to focus on active threats, perhaps because of their urgency. We must not neglect systemic features that allow error and omissions to escalate. Passive threats can also cause catastrophes. There is a need to put more emphasis on long-term countermeasures because that is where true risk management comes into play. Short-term countermeasures have been easier to define but do not look into potential threats and susceptibilities.

There also needs to be more of a look at risks not based solely on threats and assets. Inherent risks, reliability, and development risks are all factors that need to be added to the risk management considerations.

8.10 SYSTEM CHARACTERISTICS

There are some significant differences between IT systems and other engineering products that affect security risk management. Many of the techniques applicable to continuous physical systems do not apply to IT systems because of discontinuity in hardware and software (Zelkowitz, 1994). IT systems can be unforgiving and are becoming more unpredictable. Small errors, problems, or breaches can have large effects. It is not clear whether it is possible to over-engineer software systems to build in a margin for error, as it is in other engineering disciples, especially manufacturing.

8.11 CONTINGENCY PLANS

Contingency planning in the form of backup data storage and remote alternate operation sites has long been part of security planning for availability, reflecting the preparation for undesirable event occurrences. The safety community practices risk toleration as another form of contingency planning. Contingency planning needs to be extended to other security policy objectives such as confidentiality and integrity. The paradigm shift away from risk prevention implies occasional policy breaches for which provision should be made.

8.12 DISSEMINATING INFORMATION

While risk management must function in the absence of sufficient information, there is no virtue in taking on an unnecessary handicap. Accumulated information about risks and countermeasures must be made available. Disseminating such information is, itself, a risk management decision. It is not possible to keep the information from the penetrators, nor is it possible to force all system administrators to act with what we consider necessary prudence.

However, decision-makers have always been reluctant to keep and/or provide records of the risk management decisions being made. There are many reasons for this. These decisions are not made with a great deal of quantitative data and are therefore highly subjective in nature. Without objective, quantitative data to back up decisions, many responsible managers do not wish to be second-guessed years after a decision has been made. Therefore, records are seen as a liability and not diligently kept.

SECTION 9

CLOSING

The attendees of the workshop were asked three questions at the closing session: "Was there utility in this first workshop for IT security?" "Should there be future such workshops?" and "If so (to the first two questions), what should be the frequency of the workshops ?"

9.1 IT SECURITY ASSURANCE WORKSHOP UTILITY

The general consensus was that a forum to discuss the issues of IT security assurance was of great use to the community. In particular, this workshop determined that assurance is still a somewhat nebulous subject. There are many questions that need to be explored. It is still difficult to define precisely what is meant by assurance, and the definition varies from person to person and enterprise to enterprise. The questions of how to gain assurance, how to relay assurance gained, and how to use assurance all need further study. It was generally felt that just the identification of these questions for further study made the workshop a useful exercise.

There was some sentiment that these subjects need to be pushed back out into the community for actual resolution. Discussions in a short workshop must remain at a high level. Therefore, no problems could be resolved at such meetings. However, the group decided that this group could help provide the questions that could be addressed through research. One method for doing this in the near future is to provide the proceedings to other forums considering the subject of assurance, especially those attending the International Workshop of Developmental Assurance being held in Maryland in June 1994. In addition, a panel discussion on the results of both of the assurance workshops has been scheduled for the National Computer Security Conference in October 1994. It is thought that these forums will further advance the discussions and move the community one step closer to resolution of some of the issues identified.

9.2 FUTURE ASSURANCE WORKSHOPS

Based on the utility discussion, the group agreed that future workshops should take place. However, the group also agreed that it should strive to move beyond identifying issues to resolution of some of these issues. It was agreed that this could happen once the problems are well defined. Workshop sessions could then concentrate on issues of much finer detail.

9.3 FREQUENCY OF WORKSHOPS

The group did not determine a specific duration between workshops. Instead, it was generally decided that they should be planned as needed. Because of the impending Developmental Workshop and the assurance discussion planned for the NCSC in October, the planning committee has decided to target spring 1995 for the next workshop.

9.4 OBSERVATIONS ON PROGRESS

One conclusion/observation is that the security community has made little progress in the past years in truly understanding the issues at hand, and there is little hope for the immediate future. We continue to spend large amounts of resources trying to address simple issues such as definition of terms and high-level processes (again and again), while failing to understand the *user community's* needs and promote a strong security awareness among the general user population. While there appeared to be much agreement on what had been done wrong in the past, there appeared to be little consensus on how to proceed. One thing appears clear though: with the rapidly changing technology and threat environment, unless the security community becomes more proactive, more in touch with the real user needs and expectations, and does a better job of developing security awareness in the user community (to ensure security is built in *and* maintained during operation of the system), the security of our information and resources will not improve.

LIST OF REFERENCES

Bell, D. E., and L. J. LaPadula, 1975, *Secure Computer Systems: Generalized Framework for Exposition and Multics Interpretation*, ESD-TR-75-306, The MITRE Corporation. (Also available through NIST, Springfield, VA AD-A023588)

Canadian System Security Center, January 1993, *The Canadian Trusted Computer Product Evaluation Criteria*, Version 3.0e.

Commission of the European Communities, 28 June 1991, *Information Technology Security Evaluation Criteria (ITSEC): Provisional Harmonized Criteria*, Luxembourg: Office for Official Publications of the European Communities, Version 1.2.

Computer Science Laboratory, March 1992, *EHDM Specifications and Verification System - Version 6.1 User's Guide*, Technical Report, SRI International.

Department of Defense, 1985, *Department of Defense Trusted Computer System Evaluation Criteria*, DOD 5200.28-STD, Washington, DC.

European Computer Manufacturers Association, 14 December 1989, *Security in Open Systems, Data Elements and Device Definitions*, ECMA-138.

ISO 9000, International Standards for Quality Management, 2nd Edition, 1987-1991

Joint Security Committee Report, 28 February 1994, *Redefining Security: A Report to the Secretary of Defense and the Director of Central Intelligence*, Joint Security Committee, Washington DC.

Korelsky, Tanya, and David Rosenthal, August 1992, *Integrated Trusted Systems Development Environment Tools Final Report*, Technical Report, ORA Corporation.

McEnerney, Joseph R., D. G. Weber, Randall Brown, and R. Varadarajah, October 1990, "Automated Extensibility in THETA," *Proceedings 13th National Computer Security Conference*, Washington, D.C.

McDermid, John A., 1991, "Issues in Developing Software for Safety Critical Systems," *Reliability Engineering and System Safety*, Volume 32, 1991, pages 1-24.

National Computer Security Center, 25 June 1985, *Computer Security Requirements: Guidance for Applying the Department of Defense Trusted Computer System Evaluation Criteria in Specific Environments* (Yellow Books), National Security Agency, Fort George G. Meade, MD.

National Institute of Standards and Technology and National Security Agency, December 1992, *Federal Criteria for Information Technology Security*, Version 1.0.

National Research Council, 1991, *Computers at Risk Safe Computing In the Information Age*.

O'Brien, Richard, and Clyde Rogers, October 1991, "Developing Applications on LOCK," *14th National Computer Security Conference*.

Saltman, Roy G., December 1993, *Good Security Practices for Electronic Commerce, Including Electronic Data Interchange*.

Zelkowitz, M. V., 1994, *Models and Algebra (and Reality)*.

APPENDIX

ATTENDEES

Listed below are the workshop attendees (asterisks indicate members of the organizing committee):

Marshall Abrams*
The MITRE Corporation
7525 Colshire Drive
McLean, VA 22102
(703) 883-6938
abrams@mitre.org

Dee Akers*
The MITRE Corporation
7525 Colshire Drive
MS Z274
McLean, VA 22102
(703) 883-5907
akers@mitre.org

Lynn Ambuel*
NSA,I94
Fort Meade, MD 20755-600
(410) 859-4463
ambuel@dockmaster.ncsc.mil

Blaine Burnham
NSA,V12
9800 Savage Rd
Fort Meade, MD 20755-6020
burnham@dockmaster.ncsc.mil

Debbie Campbell
NSA/I94
9800 Savage Rd
Fort George G. Meade, MD 20755
(401) 859-4464
dscampbell@dockmaster.ncsc.mil

Julie Connolly
NSA/I91
9800 Savage Rd
Fort Meade, MD 20755
(410) 684-7374
jlconnolly@dockmaster.ncsc.mil

Karen Ferraiolo
ARCA Systems, Inc.
8229 Boone Blvd., #610
Vienna, VA 22182
(703) 724-5611
ferraiolo@arca.va.com

Sharon Fletcher
Sandia National Laboratories
Dept. 9411
Albuquerque, NM 87185-0778
(505) 844-2251
skflete@sandia.gov

Lester Fraim
The MITRE Corporation
7525 Colshire Drive
McLean, VA 22102
(703) 883-6339
fraim@mitre.org

Art Friedman
The MITRE Corporation
7525 Colshire Drive
McLean, VA 22102
(410) 859-6700/01
arf@mitre.org

Judith N. Froscher*
Naval Research Laboratory
Code 5542
Washington, DC 20375
(202) 767-3012
froscher@itd.nrl.navy.mil

Bret Hartman
Odyssey Research Associates
301 Dates Drive
Ithaca, NY 14850
(607) 277-2020
bret@oracorp.com

Chuck Howell
The MITRE Corporation
7525 Colshire Drive
McLean, VA 22102
(703) 883-6080
howell@mitre.org

Jay Kahn
The MITRE Corporation
7525 Colshire Drive
McLean, VA 22102-3481
(703) 883-6622
jkahn@mitre.org

Doug Landall
ARCA Systems
10320 Little Patuxent Parkway
Columbia, MD 21044
(410) 715-0500
landoll.arca.md.com

Burkhard Lau
Delft University of Technology
P.O. Box 365
NL-2624 AJ Delft
Netherlands
31-15-787106
B.Lau@IS.TWI.TUDelft.NL

John McDermid
University of York
Heslington, York
Y015DD, UK
44 904 432726
jam@minster.york.ac.uk

Piers McMahon
ICL
Kings Road
33 Kings Road
Reading, RG13PK, UK
pvmcmahon@rea0803.mins.icl.co.uk
+44 734 634882

Bill Neugent
The MITRE Corporation
7525 Colshire Drive
McLean, VA 22102
(703) 883-6632
wneugent@smiley.mitre.org

Ingrid Olson
The MITRE Corporation
7525 Colshire Drive
McLean, VA 22102
(703) 883-7044
iolson@smiley.mitre.org

Bernard Roussely
SCSSI
18, rue du Docteur Zamenhof
G2131 ISSY-LES-MOULINEAUX
France
SSI13@calvacom.fr

Joel Sachs
ARCA Systems
One Commerce Center
10320 Little Patuxent Pkwy
Suite 1005
Columbia, MD 21044
(410) 715-0500
sachs@area.md.com

Ravi Sandhu
George Mason University
Issue Department, MS4A4
Fairfax, VA 22030
(703) 993-1659
sandhu@gmu.edu

Dan Sterne
Trusted Information Systems
3060 Washington Road
Glenwood, MD 21738
(301) 854-6889
sterne@tis.com

Pat Toth*
National Institute of Standards and
Technology
National Computer Systems Laboratory
Gaithersburg, MD 20899
(302) 975-5140
toth@csmes.ncsl.nist.gov

Caralyn Wichers*
BBN
9810 Patuxent Woods Drive
Columbia, MD 21046-1562
(410) 290-6188
cwichers@bbn.com

Jim Williams
The MITRE Corporation
202 Burlington Rd
Bedford, MA 01730
(617) 271-2647
jgw@mitre.org

GLOSSARY

ACSA	Annual Computer Security Applications Conference
ADP	automatic data processing
ANSI	American National Standards Institute
ARPA	Advanced Research Projects Agency
CAS	Controlled Application Set
COTS	commercial off-the-shelf
DOD	Department of Defense
ECMA	European Computer Manufacturers Association
INFOSEC	information security
ISO	International Standards Organization
IT	information technology
IWITAT	Invitational Workshop on Information Technology Assurance and Trustworthiness
JSC	Joint Security Commission
MAC	mandatory access control
MLS	multilevel secure
NIST	National Institute of Standards and Technology
NRL	Naval Research Laboratory
NSA	National Security Agency

OO	object oriented
OS	operating systems
R&D	research and development
SINTRA	Secure Information Through Replicated Architecture
SQL	Structured Query Language
TCB	trusted computing base
TCSEC	Trusted Computer System Evaluation Criteria
U.S.	United States