8.4.1.1 Conducting a Sensitivity Assessment

The definition of *sensitive* is often misconstrued. *Sensitive* is synonymous with *important* or *valuable*. Some data is sensitive because it must be kept confidential. Much more data, however, is sensitive because its integrity or availability must be assured. The Computer Security Act and OMB Circular A-130 clearly state that information is sensitive if its unauthorized disclosure, modification (i.e., loss of integrity), or unavailability would harm the agency. In general, the more important a system is to the mission of the agency, the more sensitive it is.

A *sensitivity assessment* looks at the sensitivity of both the information to be processed and the system itself. The assessment should consider legal implications, organization policy (including federal and agency policy if a federal system), and the functional needs of the system. Sensitivity is normally expressed in terms of integrity, availability, and confidentiality. Such factors as the importance of the system to the organization's mission and the consequences of unauthorized modification, unauthorized disclosure, or unavailability of the system or data need to be examined when assessing sensitivity. To address these types of issues, the people who use or own the system or information should participate in the assessment.

A sensitivity assessment should answer the following questions:

- What information is handled by the system?
- What kind of potential damage could occur through error, unauthorized disclosure or modification, or unavailability of data or the system?
- What laws or regulations affect security (e.g., the Privacy Act or the Fair Trade Practices Act)?
- To what threats is the system or information particularly vulnerable?
- Are there significant environmental considerations (e.g., hazardous location of system)?
- What are the security-relevant characteristics of the user community (e.g., level of technical sophistication and training or security clearances)?
- What internal security standards, regulations, or guidelines apply to this system?

The sensitivity assessment starts an analysis of security that continues throughout the life cycle. The assessment helps determine if the project needs special security oversight, if further analysis is

needed before committing to begin system development (to ensure feasibility at a reasonable cost), or in rare instances, whether the security requirements are so strenuous and costly that system development or acquisition will not be pursued. The sensitivity assessment can be included with the system initiation documentation either as a separate document or as a section of another planning document. The development of security features, procedures, and assurances, described in the next section, builds on the sensitivity assessment.

A sensitivity assessment can also be performed during the planning stages of system upgrades (for either upgrades being procured or developed in house). In this case, the assessment focuses on the affected areas. If the upgrade significantly affects the original assessment, steps can be taken to analyze the impact on the rest of the system. For example, are new controls needed? Will some controls become unnecessary?

8.4.2 Development/Acquisition

For most systems, the development/acquisition phase is more complicated than the initiation phase. Security activities can be divided into three parts:

- determining security features, assurances, and operational practices;
- incorporating these security requirements into design specifications; and
- actually acquiring them.

These divisions apply to systems that are designed and built in house, to systems that are purchased, and to systems developed using a hybrid approach.

During this phase, technical staff and system sponsors should actively work together to ensure that the technical designs reflect the system's security needs. As with development and incorporation of other system requirements, this process requires an open dialogue between technical staff and system sponsors. It is important to address security requirements effectively in synchronization with development of the overall system.

8.4.2.1 Determining Security Requirements

During the first part of the development/ acquisition phase, system planners define the requirements of the system. *Security requirements should be developed at the same time.* These requirements can be expressed as technical features (e.g., access controls), assurances (e.g., background checks for system developers), or operational practices (e.g., awareness and training). System security requirements, like other system requirements, are derived from a number of sources including law, policy, applicable standards and guidelines, functional needs of the system, and cost-benefit trade-offs.

Law. Besides specific laws that place security requirements on information, such as the Privacy Act of 1974, there are laws, court cases, legal opinions, and other similar legal material that may affect security directly or indirectly.

Policy. As discussed in Chapter 5, management officials issue several different types of policy. System security requirements are often derived from issue-specific policy.

Standards and Guidelines. International, national, and organizational standards and guidelines are another source for determining security features, assurances, and operational practices. Standards and guidelines are often written in an "if...then" manner (e.g., if the system is encrypting data, then a particular cryptographic algorithm should be used). Many organizations specify baseline controls for different types of systems, such as administrative, mission- or business-critical, or proprietary. As required, special care should be given to interoperability standards.

Functional Needs of the System. The purpose of security is to support the function of the system, not to undermine it. Therefore, many aspects of the function of the system will produce related security requirements.

Cost-Benefit Analysis. When considering security, cost-benefit analysis is done through risk assessment, which examines the assets, threats, and vulnerabilities of the system in order to determine the most appropriate, cost-effective safeguards (that comply with applicable laws, policy, standards, and the functional needs of the system). Appropriate safeguards are normally those whose anticipated benefits outweigh their costs. Benefits and costs include monetary and nonmonetary issues, such as prevented losses, maintaining an organization's reputation, decreased user friendliness, or increased system administration.

Risk assessment, like cost-benefit analysis, is used to support decision making. It helps managers select cost-effective safeguards. The extent of the risk assessment, like that of other cost-benefit analyses, should be commensurate with the complexity and cost (normally an indicator of complexity) of the system and the expected benefits *of the assessment*. Risk assessment is further discussed in Chapter 7.

Risk assessment can be performed during the requirements analysis phase of a procurement or the design phase of a system development cycle. Risk should also normally be assessed during the development/acquisition phase of a system upgrade. The risk assessment may be performed once or multiple times, depending upon the project's methodology.

Care should be taken in differentiating between *security* risk assessment and *project* risk analysis. Many system development and acquisition projects analyze the risk of failing to successfully complete the project – a different activity from *security* risk assessment.

8.4.2.2 Incorporating Security Requirements Into Specifications

Determining security features, assurances, and operational practices can yield significant security information and often voluminous requirements. This information needs to be validated, updated, and organized into the detailed security protection requirements and specifications used by systems designers or purchasers. Specifications can take on quite different forms, depending on the methodology used for to develop the system, or whether the system, or parts of the system, are being purchased off the shelf.

As specifications are developed, it may be necessary to update initial risk assessments. A safeguard recommended by the risk assessment could be incompatible with other requirements, or a control may be difficult to implement. For example, a security requirement that prohibits dial-in access could prevent employees from checking their e-mail while away from the office.[69]

> Developing testing specifications early can be critical to being able to cost-effectively test security features.

Besides the technical and operational controls of a system, assurance also should be addressed. The degree to which assurance (that the security features and practices can and do work correctly and effectively) is needed should be determined early. Once the desired level of assurance is determined, it is necessary to figure out how the system will be tested or reviewed to determine whether the specifications have been satisfied (to obtain the desired assurance). This applies to both system developments and acquisitions. For example, if rigorous assurance is needed, the ability to test the system or to provide another form of initial and ongoing assurance needs to be designed into the system or otherwise provided for. See Chapter 9 for more information.

8.4.2.3 Obtaining the System and Related Security Activities

During this phase, the system is actually built or bought. If the system is being built, security activities may include developing the system's security aspects, monitoring the development process itself for security problems, responding to changes, and monitoring threat. Threats or vulnerabilities that may arise during the development phase include Trojan horses, incorrect code, poorly functioning development tools, manipulation of code, and malicious insiders.

If the system is being acquired off the shelf, security activities may include monitoring to ensure security is a part of market surveys, contract solicitation documents, and evaluation of proposed systems. Many systems use a combination of development and acquisition. In this case, security activities include both sets.

[69] This is an example of a risk-based decision.

As the system is built or bought, choices are made about the system, which can affect security. These choices include selection of specific off-the-shelf products, finalizing an architecture, or selecting a processing site or platform. Additional security analysis will probably be necessary.

In federal government contracting, it is often useful if personnel with security expertise participate as members of the source selection board to help evaluate the security aspects of proposals.

In addition to obtaining the system, operational practices need to be developed. These refer to human activities that take place around the system such as contingency planning, awareness and training, and preparing documentation. The chapters in the Operational Controls section of this handbook discuss these areas. These need to be developed along with the system, although they are often developed by different individuals. These areas, like technical specifications, should be considered from the beginning of the development and acquisition phase.

8.4.3 Implementation

A separate implementation phase is not always specified in some life cycle planning efforts. (It is often incorporated into the end of development and acquisition or the beginning of operation and maintenance.) However, from a security point of view, a critical security activity, *accreditation*, occurs between development and the start of system operation. The other activities described in this section, turning on the controls and testing, are often incorporated at the end of the development/acquisition phase.

8.4.3.1 Install/Turn-On Controls

While obvious, this activity is often overlooked. When acquired, a system often comes with security features disabled. These need to be enabled and configured. For many systems this is a complex task requiring significant skills. Custom-developed systems may also require similar work.

8.4.3.2 Security Testing

System security testing includes both the testing of the particular parts of the system that have been developed or acquired and the testing of the entire system. Security management, physical facilities, personnel, procedures, the use of commercial or in-house services (such as networking services), and contingency planning are examples of areas that affect the security of the entire system, but may be specified outside of the development or acquisition cycle. Since only items within the development or acquisition cycle will have been tested during system acceptance testing, separate tests or reviews may need to be performed for these additional security elements.

Security certification is a formal testing of the security safeguards implemented in the computer system to determine whether they meet applicable requirements and specifications.[70] To provide more reliable technical information, certification is often performed by an independent reviewer, rather than by the people who designed the system.

8.4.3.3 Accreditation

System security accreditation is the *formal authorization* by the accrediting (management) official for system operation and an *explicit acceptance of risk*. It is usually supported by a review of the system, including its management, operational, and technical controls. This review may include a detailed technical evaluation (such as a Federal Information Processing Standard 102 certification, particularly for complex, critical, or high-risk systems), security evaluation, risk assessment, audit, or other such review. If the life cycle process is being used to manage a project (such as a system upgrade), it is important to recognize that the accreditation is for the entire system, not just for the new addition.

The best way to view computer security accreditation is as a form of quality control. It forces managers and technical staff to work together to find the best fit for security, given technical constraints, operational constraints, and mission requirements. The accreditation process obliges managers to make critical decisions regarding the adequacy of security safeguards. A decision based on reliable information about the effectiveness of technical and non-technical safeguards and the residual risk is more likely to be a sound decision.

Sample Accreditation Statement

In accordance with (Organization Directive), I hereby issue an accreditation for (name of system). This accreditation is my formal declaration that a satisfactory level of operational security is present and that the system can operate under reasonable risk. This accreditation is valid for three years. The system will be re-evaluated annually to determine if changes have occurred affecting its security.

After deciding on the acceptability of security safeguards and residual risks, the accrediting official should issue a formal accreditation statement. While most flaws in system security are not severe enough to remove an operational system from service or to prevent a new system from becoming operational, the flaws may require some restrictions on operation (e.g., limitations on dial-in access or electronic connections to other organizations). In some cases, an interim accreditation may be granted, allowing the system to operate requiring review at the end of the

[70] Some federal agencies use a broader definition of the term certification to refer to security reviews or evaluations, formal or information, that take place prior to and are used to support accreditation.

interim period, presumably after security upgrades have been made.

8.4.4 Operation and Maintenance

Many security activities take place during the operational phase of a system's life. In general, these fall into three areas: (1) security operations and administration; (2) operational assurance; and (3) periodic re-analysis of the security. Figure 8.2 diagrams the flow of security activities during the operational phase.

8.4.4.1 Security Operations and Administration

Operation of a system involves many security activities discussed throughout this handbook. Performing backups, holding training classes, managing cryptographic keys, keeping up with user administration and access privileges, and updating security software are some examples.

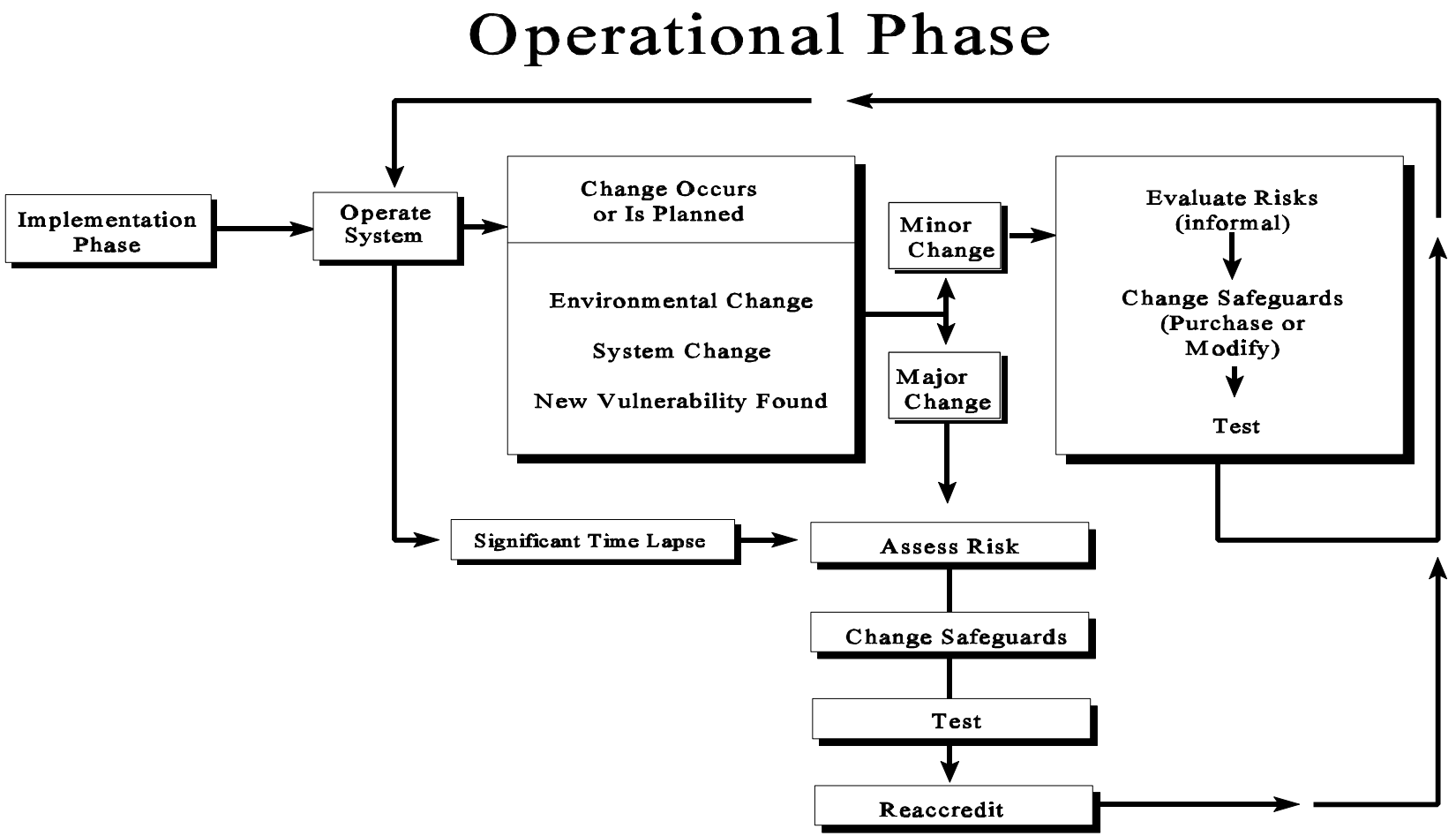
8.4.4.2 Operational Assurance

> Operational assurance examines whether a system is operated according to its current security requirements. This includes both the actions of people who operate or use the system and the functioning of technical controls.

Security is *never* perfect when a system is implemented. In addition, system users and operators discover new ways to intentionally or unintentionally bypass or subvert security. Changes in the system or the environment can create new vulnerabilities. Strict adherence to procedures is rare over time, and procedures become outdated. Thinking risk is minimal, users may tend to bypass security measures and procedures.

As shown in Figure 8.2, changes occur. Operational assurance is one way of becoming aware of these changes whether they are new vulnerabilities (or old vulnerabilities that have not been corrected), system changes, or environmental changes. Operational assurance is the process of reviewing an operational system to see that security controls, both automated and manual, are functioning correctly and effectively.

To maintain operational assurance, organizations use two basic methods: *system audits* and *monitoring*. These terms are used loosely within the computer security community and often overlap. A system audit is a one-time or periodic event to evaluate security. Monitoring refers to an ongoing activity that examines either the system or the users. In general, the more "real-time" an activity is, the more it falls into the category of monitoring. (See Chapter 9.)



During the operational phase of a system life cycle, major and minor changes will occur. This figure diagrams appropriate responses to change to help ensure the continued security of the system at a level acceptable to the accrediting official.

Figure 8.2

8.4.4.3 Managing Change

Security change management helps develop new security requirements.

Computer systems and the environments in which they operate change continually. In response to various events such as user complaints, availability of new features and services, or the discovery of new threats and vulnerabilities, system managers and users modify the system and incorporate new features, new procedures, and software updates.

The environment in which the system operates also changes. Networking and interconnections tend to increase. A new user group may be added, possibly external groups or anonymous groups. New threats may emerge, such as increases in network intrusions or the spread of personal computer viruses. If the system has a configuration control board or other structure to manage technical system changes, a security specialist can be assigned to the board to make determinations about whether (and if so, how) changes will affect security.

Security should also be considered during system upgrades (and other planned changes) and in determining the impact of unplanned changes. As shown in Figure 8.2, when a change occurs or is planned, a determination is made whether the change is major or minor. A major change, such as reengineering the structure of the system, significantly affects the system. Major changes often involve the purchase of new hardware, software, or services or the development of new software modules.

An organization does not need to have a specific cutoff for major-minor change decisions. A sliding scale between the two can be implemented by using a combination of the following methods:

- *Major change*. A major change requires analysis to determine security requirements. The process described above can be used, although the analysis may focus only on the area(s) in which the change has occurred or will occur. If the original analysis and system changes have been documented throughout the life cycle, the analysis will normally be much easier. Since these changes result in significant system acquisitions, development work, or changes in policy, the system should be reaccredited to ensure that the residual risk is still acceptable.

- *Minor change*. Many of the changes made to a system do not require the extensive analysis performed for major changes, but do require some analysis. Each change can involve a limited risk assessment that weighs the pros (benefits) and cons (costs) and that can even be performed on-the-fly at meetings. Even if the analysis is conducted informally, decisions should still be appropriately documented. This process recognizes that even "small" decisions should be

risk-based.

8.4.4.4 Periodic Reaccreditation

Periodically, it is useful to formally reexamine the security of a system from a wider perspective. The analysis, which leads to reaccreditation, should address such questions as: Is the security still sufficient? Are major changes needed?

The reaccreditation should address high-level security and management concerns as well as the implementation of the security. It is not always necessary to perform a new risk assessment or certification in conjunction with the re-accreditation, but the activities support each other (and both need be performed periodically). The more extensive system changes have been, the more extensive the analyses should be (e.g., a risk assessment or re-certification). A risk assessment is likely to uncover security concerns that result in system changes. After the system has been changed, it may need testing (including certification). Management then reaccredits the system for continued operation if the risk is acceptable.

> It is important to consider legal requirements for records retention when disposing of computer systems. For federal systems, system management officials should consult with their agency office responsible for retaining and archiving federal records.

8.4.5 Disposal

The disposal phase of the computer system life cycle involves the disposition of information, hardware, and software. Information may be moved to another system, archived, discarded, or destroyed. When archiving information, consider the method for retrieving the information in the future. The technology used to create the records may not be readily available in the future.

Hardware and software can be sold, given away, or discarded. There is rarely a need to destroy hardware, except for some storage media containing confidential information that cannot be sanitized without destruction. The disposition of software needs to be in keeping with its license or other agreements with the developer, if applicable. Some licenses are

> **Media Sanitization**
>
> Since electronic information is easy to copy and transmit, information that is sensitive to disclosure often needs to be controlled throughout the computer system life cycle so that managers can ensure its proper disposition. The removal of information from a storage medium (such as a hard disk or tape) is called *sanitization*. Different kinds of sanitization provide different levels of protection. A distinction can be made between clearing information (rendering it unrecoverable by keyboard attack) and purging (rendering information unrecoverable against laboratory attack). There are three general methods of purging media: overwriting, degaussing (for magnetic media only), and destruction.

site-specific or contain other agreements that prevent the software from being transferred.

Measures may also have to be taken for the future use of data that has been encrypted, such as taking appropriate steps to ensure the secure long-term storage of cryptographic keys.

8.5 Interdependencies

Like many management controls, life cycle planning relies upon other controls. Three closely linked control areas are policy, assurance, and risk management.

Policy. The development of system-specific policy is an integral part of determining the security requirements.

Assurance. Good life cycle management provides assurance that security is appropriately considered in system design and operation.

Risk Management. The maintenance of security throughout the operational phase of a system is a process of risk management: analyzing risk, reducing risk, and monitoring safeguards. Risk assessment is a critical element in designing the security of systems and in reaccreditations.

8.6 Cost Considerations

Security is a factor throughout the life cycle of a system. Sometimes security choices are made by default, without anyone analyzing why choices are made; sometimes security choices are made carefully, based on analysis. The first case is likely to result in a system with poor security that is susceptible to many types of loss. In the second case, the cost of life cycle management should be *much smaller* than the losses avoided. The major cost considerations for life cycle management are personnel costs and some delays as the system progresses through the life cycle for completing analyses and reviews and obtaining management approvals.

It is possible to overmanage a system: to spend more time planning, designing, and analyzing risk than is necessary. Planning, by itself, does not further the mission or business of an organization. Therefore, while security life cycle management can yield significant benefits, the effort should be commensurate with the system's size, complexity, and sensitivity and the risks associated with the system. In general, the higher the value of the system, the newer the system's architecture, technologies, and practices, and the worse the impact if the system security fails, the more effort should be spent on life cycle management.

References

Communications Security Establishment. *A Framework for Security Risk Management in*

Information Technology Systems. Canada.

Dykman, Charlene A. ed., and Charles K. Davis, asc. ed. *Control Objectives – Controls in an Information Systems Environment: Objectives, Guidelines, and Audit Procedures*. (fourth edition). Carol Stream, IL: The EDP Auditors Foundation, Inc., April 1992.

Guttman, Barbara. *Computer Security Considerations in Federal Procurements: A Guide for Procurement Initiators, Contracting Officers, and Computer Security Officials*. Special Publication 800-4. Gaithersburg, MD: National Institute of Standards and Technology, March 1992.

Institute of Internal Auditors Research Foundation. *System Auditability and Control Report*. Altamonte Springs, FL: The Institute of Internal Auditors, 1991.

Murphy, Michael, and Xenia Ley Parker. *Handbook of EDP Auditing*, especially Chapter 2 "The Auditing Profession," and Chapter 3, "The EDP Auditing Profession." Boston, MA: Warren, Gorham & Lamont, 1989.

National Bureau of Standards. *Guideline for Computer Security Certification and Accreditation*. Federal Information Processing Standard Publication 102. September 1983.

National Institute of Standards and Technology. "Disposition of Sensitive Automated Information." Computer Systems Laboratory Bulletin. October 1992.

National Institute of Standards and Technology. "Sensitivity of Information." Computer Systems Laboratory Bulletin. November 1992.

Office of Management and Budget. "Guidance for Preparation of Security Plans for Federal Computer Systems That Contain Sensitive Information." OMB Bulletin 90-08. 1990.

Ruthberg, Zella G, Bonnie T. Fisher and John W. Lainhart IV. *System Development Auditor*. Oxford, England: Elsevier Advanced Technology, 1991.

Ruthberg, Z., et al. *Guide to Auditing for Controls and Security: A System Development Life Cycle Approach*. Special Publication 500-153. Gaithersburg, MD: National Bureau of Standards. April 1988.

Vickers Benzel, T. C. *Developing Trusted Systems Using DOD-STD-2167A*. Oakland, CA: IEEE Computer Society Press, 1990.

Wood, C. "Building Security Into Your System Reduces the Risk of a Breach." *LAN Times*, 10(3), 1993. p 47.

Chapter 9

ASSURANCE

Computer security assurance is the degree of confidence one has that the security measures, both technical and operational, work as intended to protect the system and the information it processes. Assurance is not, however, an absolute guarantee that the measures work as intended. Like the closely related areas of reliability and quality, assurance can be difficult to analyze; however, it is something people expect and obtain (though often without realizing it). For example, people may routinely get product recommendations from colleagues but may not consider such recommendations as providing assurance.

Assurance is a degree of confidence, not a true measure of how secure the system actually is. This distinction is necessary because it is extremely difficult -- and in many cases virtually impossible -- to know exactly how secure a system is.

> Security assurance is the degree of confidence one has that the security controls operate correctly and protect the system as intended.

Assurance is a challenging subject because it is difficult to describe and even more difficult to quantify. Because of this, many people refer to assurance as a "warm fuzzy feeling" that controls work as intended. However, it is possible to apply a more rigorous approach by knowing two things: (1) who needs to be assured and (2) what types of assurance can be obtained. The person who needs to be assured is the management official who is ultimately responsible for the security of the system. Within the federal government, this person is the *authorizing or accrediting official*.[71]

There are many methods and tools for obtaining assurance. For discussion purposes, this chapter categorizes assurance in terms of a general system life cycle. The chapter first discusses planning for assurance and then presents the two categories of assurance methods and tools: (1) design and implementation assurance and (2) operational assurance. Operational assurance is further categorized into audits and monitoring.

The division between design and implementation assurance and operational assurance can be fuzzy. While such issues as configuration management or audits are discussed under operational assurance, they may also be vital during a system's development. The discussion tends to focus more on technical issues during design and implementation assurance and to be a mixture of

[71] Accreditation is a process used primarily within the federal government. It is the process of managerial authorization for processing. Different agencies may use other terms for this approval function. The terms used here are consistent with Federal Information Processing Standard 102, *Guideline for Computer Security Certification and Accreditation*. (See reference section of this chapter.)

management, operational, and technical issues under operational assurance. The reader should keep in mind that the division is somewhat artificial and that there is substantial overlap.

9.1 Accreditation and Assurance

Accreditation is a management official's formal acceptance of the adequacy of a system's security. The best way to view computer security accreditation is as a form of quality control. It forces managers and technical staff to work together to find workable, cost-effective solutions given security needs, technical constraints, operational constraints, and mission or business requirements. The accreditation process obliges managers to make the critical decision regarding the adequacy of security safeguards and, therefore, to recognize and perform their role in securing their systems. In order for the decisions to be sound, they need to be based on reliable information about the implementation of both technical and nontechnical safeguards. These include:

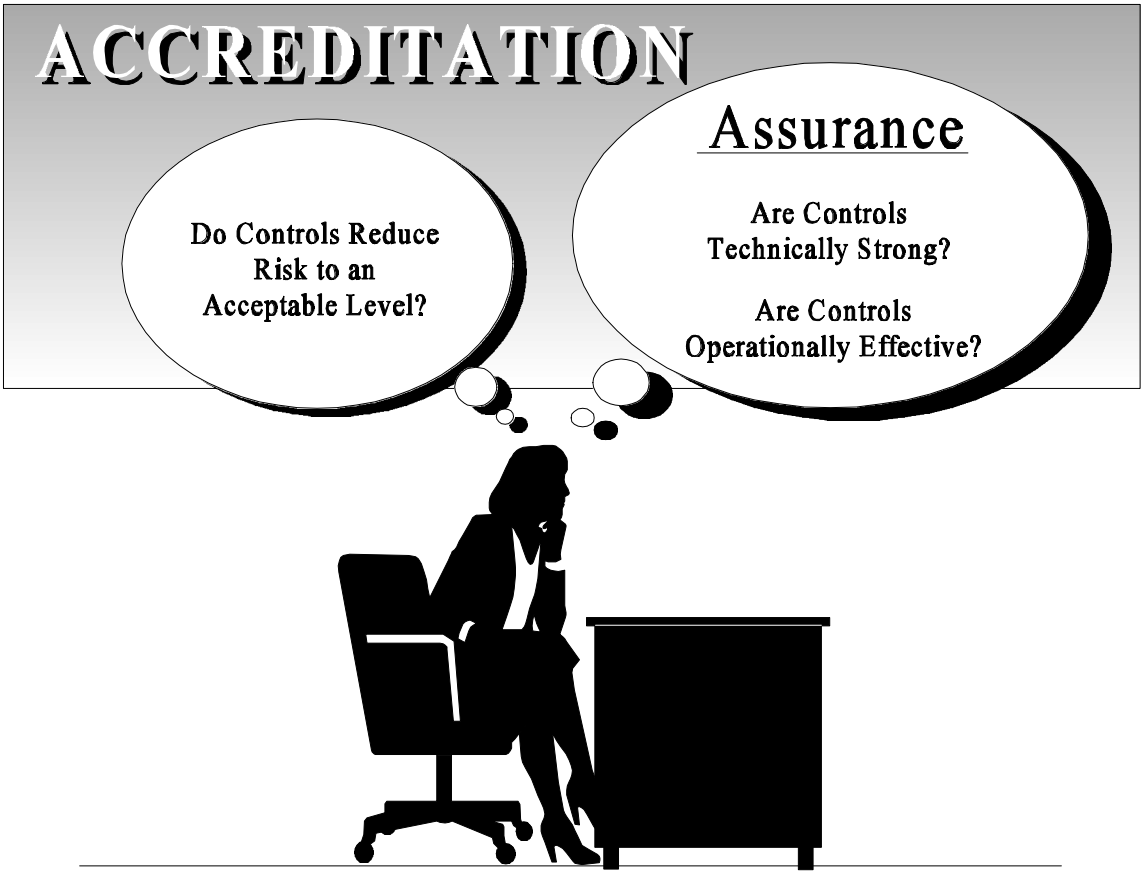
- Technical features (Do they operate as intended?).
- Operational practices (Is the system operated according to stated procedures?).
- Overall security (Are there threats which the technical features and operational practices do not address?).
- Remaining risks (Are they acceptable?).

A computer system should be accredited before the system becomes operational with periodic reaccreditation after major system changes or when significant time has elapsed.[72] Even if a system was not initially accredited, the accreditation process can be initiated at any time. Chapter 8 further discusses accreditation.

9.1.1 Accreditation and Assurance

Assurance is an extremely important -- but not the only -- element in accreditation. As shown in the diagram, assurance addresses whether the technical measures and procedures operate either (1) according to a set of security requirements and specifications or (2) according to general quality principles. Accreditation also addresses whether the system's security requirements are correct and well implemented and whether the level of quality is sufficiently high. These activities are discussed in Chapters 7 and 8.

[72] OMB Circular A-130 requires management security authorization of operation for federal systems.



9.1.2 Selecting Assurance Methods

The accrediting official makes the final decision about how much and what types of assurance are needed for a system. For this decision to be informed, it is derived from a review of security, such as a risk assessment or other study (e.g., certification), as deemed appropriate by the accrediting official.[73] The accrediting official needs to be in a position to analyze the pros and cons of the cost of assurance, the cost of controls, and the risks to the organization. At the end of the accreditation process, the accrediting official will be the one to accept the remaining risk. Thus,

[73] In the past, accreditation has been defined to require a certification, which is an in-depth testing of technical controls. It is now recognized within the federal government that other analyses (e.g., a risk analysis or audit) can also provide sufficient assurance for accreditation.

the selection of assurance methods should be coordinated with the accrediting official.

In selecting assurance methods, the need for assurance should be weighed against its cost. Assurance can be quite expensive, especially if extensive testing is done. Each method has strengths and weaknesses in terms of cost and what kind of assurance is actually being delivered. A combination of methods can often provide greater assurance, since no method is foolproof, and can be less costly than extensive testing.

The accrediting official is not the only arbiter of assurance. Other officials who use the system should also be consulted. (For example, a Production Manager who relies on a Supply System should provide input to the Supply Manager.) In addition, there may be constraints outside the accrediting official's control that also affect the selection of methods. For instance, some of the methods may unduly restrict competition in acquisitions of federal information processing resources or may be contrary to the organization's privacy policies. Certain assurance methods may be required by organizational policy or directive.

9.2 Planning and Assurance

Assurance planning should begin during the planning phase of the system life cycle, either for new systems or a system upgrades. Planning for assurance when planning for other system requirements makes sense. If a system is going to need extensive testing, it should be built to facilitate such testing.

Planning for assurance helps a manager make decisions about what kind of assurance will be cost-effective. If a manager waits until a system is built or bought to consider assurance, the number of ways to obtain assurance may be much smaller than if the manager had planned for it earlier, and the remaining assurance options may be more expensive.

9.3 Design and Implementation Assurance

Design and implementation assurance should be examined from two points of view: the component and the system. Component assurance looks at the security of a specific product or system component, such as an operating system, application, security add-on, or telecommunications module. System assurance looks at the security of the entire system, including the interaction between products and modules.

Design and implementation assurance addresses whether the features of a system, application, or component meets security requirements and specifications and whether they are they are well designed and well built. Chapter 8 discusses the source for security requirements and specifications. Design and implementation assurance examines system design, development, and installation. Design and implementation assurance is usually associated

with the development/acquisition and implementation phase of the system life cycle; however, it should also be considered throughout the life cycle as the system is modified.

As stated earlier, assurance can address whether the product or system meets a set of security specifications, or it can provide other evidence of quality. This section outlines the major methods for obtaining design and implementation assurance.

9.3.1 Testing and Certification

Testing can address the quality of the system as built, as implemented, or as operated. Thus, it can be performed throughout the development cycle, after system installation, and throughout its operational phase. Some common testing techniques include functional testing (to see if a given function works according to its requirements) or penetration testing (to see if security can be bypassed). These techniques can range from trying several test cases to in-depth studies using metrics, automated tools, or multiple detailed test cases.

Certification is a formal process for testing components or systems against a specified set of security requirements. Certification is normally performed by an independent reviewer, rather than one involved in building the system. Certification is more often cost-effective for complex or high-risk systems. Less formal security testing can be used for lower-risk systems. Certification can be performed at many stages of the system design and implementation process and can take place in a laboratory, operating environment, or both.

9.3.2 NIST Conformance Testing and Validation Suites

NIST produces validation suites and conformance testing to determine if a product (software, hardware, firmware) meets specified standards. These test suites are developed for specific standards and use many methods. Conformance to standards can be important for many reasons, including interoperability or strength of security provided. NIST publishes a list of validated products quarterly.

9.3.3 Use of Advanced or Trusted Development

In the development of both commercial off-the-shelf products and more customized systems, the use of advanced or trusted system architectures, development methodologies, or software engineering techniques can provide assurance. Examples include security design and development reviews, formal modeling, mathematical proofs, ISO 9000 quality techniques, or use of security architecture concepts, such as a trusted computing base (TCB) or reference monitor.

9.3.4 Use of Reliable Architectures

Some system architectures are intrinsically more reliable, such as systems that use fault-tolerance,

redundance, shadowing, or redundant array of inexpensive disks (RAID) features. These examples are primarily associated with system availability.

9.3.5 Use of Reliable Security

One factor in reliable security is the concept of *ease of safe use*, which postulates that a system that is easier to secure will be more likely to be secure. Security features may be more likely to be used when the initial system defaults to the "most secure" option. In addition, a system's security may be deemed more reliable if it does not use very new technology that has not been tested in the "real" world (often called "bleeding-edge" technology). Conversely, a system that uses older, well-tested software may be less likely to contain bugs.

9.3.6 Evaluations

A product evaluation normally includes testing. Evaluations can be performed by many types of organizations, including government agencies, both domestic and foreign; independent organizations, such as trade and professional organizations; other vendors or commercial groups; or individual users or user consortia. Product reviews in trade literature are a form of evaluation, as are more formal reviews made against specific criteria. Important factors for using evaluations are the degree of independence of the evaluating group, whether the evaluation criteria reflect needed security features, the rigor of the testing, the testing environment, the age of the evaluation, the competence of the evaluating organization, and the limitations placed on the evaluations by the evaluating group (e.g., assumptions about the threat or operating environment).

9.3.7 Assurance Documentation

The ability to describe security requirements and how they were met can reflect the degree to which a system or product designer understands applicable security issues. Without a good understanding of the requirements, it is not likely that the designer will be able to meet them.

Assurance documentation can address the security either for a system or for specific components. System-level documentation should describe the system's security requirements and how they have been implemented, including *interrelationships* among applications, the operating system, or networks. System-level documentation addresses more than just the operating system, the security system, and applications; it describes the system as *integrated* and *implemented in a particular environment*. Component documentation will generally be an off-the-shelf product, whereas the system designer or implementer will generally develop system documentation.

9.3.8 Accreditation of Product to Operate in Similar Situation

The accreditation of a product or system to operate in a similar situation can be used to provide

some assurance. However, it is important to realize that an accreditation is environment- and system-specific. Since accreditation balances risk against advantages, the same product may be appropriately accredited for one environment but not for another, even by the same accrediting official.

9.3.9 Self-Certification

A vendor's, integrator's, or system developer's self-certification does not rely on an impartial or independent agent to perform a technical evaluation of a system to see how well it meets a stated security requirement. Even though it is not impartial, it can still provide assurance. The self-certifier's reputation is on the line, and a resulting certification report can be read to determine whether the security requirement was defined and whether a meaningful review was performed.

A hybrid certification is possible where the work is performed under the auspices or review of an independent organization by having that organization analyze the resulting report, perform spot checks, or perform other oversight. This method may be able to combine the lower cost and greater speed of a self-certification with the impartiality of an independent review. The review, however, may not be as thorough as independent evaluation or testing.

9.3.10 Warranties, Integrity Statements, and Liabilities

Warranties are another source of assurance. If a manufacturer, producer, system developer, or integrator is willing to correct errors within certain time frames or by the next release, this should give the system manager a sense of commitment to the product and of the product's quality. An integrity statement is a formal declaration or certification of the product. It can be backed up by a promise to (a) fix the item (warranty) or (b) pay for losses (liability) if the product does not conform to the integrity statement.

9.3.11 Manufacturer's Published Assertions

A manufacturer's or developer's published assertion or formal declaration provides a limited amount of assurance based exclusively on reputation.

9.3.12 Distribution Assurance

It is often important to know that software has arrived unmodified, especially if it is distributed electronically. In such cases, checkbits or digital signatures can provide high assurance that code has not been modified. Anti-virus software can be used to check software that comes from sources with unknown reliability (such as a bulletin board).

9.4 Operational Assurance

Design and implementation assurance addresses the quality of security features built into systems. Operational assurance addresses whether the system's technical features are being bypassed or have vulnerabilities and whether required procedures are being followed. It does not address changes in the system's security requirements, which could be caused by changes to the system and its operating or threat environment. (These changes are addressed in Chapter 8.)

Security tends to degrade during the operational phase of the system life cycle. System users and operators discover new ways to intentionally or unintentionally bypass or subvert security (especially if there is a perception that bypassing security improves functionality). Users and administrators often think that nothing will happen to them or their system, so they shortcut security. Strict adherence to procedures is rare, and they become outdated, and errors in the system's administration commonly occur.

Organizations use two basic methods to maintain operational assurance:

- A *system audit* -- a *one-time* or *periodic* event to evaluate security. An audit can vary widely in scope: it may examine an entire system for the purpose of reaccreditation or it may investigate a single anomalous event.

- *Monitoring* -- an *ongoing* activity that checks on the system, its users, or the environment.

In general, the more "real-time" an activity is, the more it falls into the category of monitoring. This distinction can create some unnecessary linguistic hairsplitting, especially concerning system-generated audit trails. Daily or weekly reviewing of the audit trail (for unauthorized access attempts) is generally monitoring, while an historical review of several months' worth of the trail (tracing the actions of a specific user) is probably an audit.

9.4.1 Audit Methods and Tools

An audit conducted to support operational assurance examines whether the system is meeting stated or implied security requirements including system and organization policies. Some audits also examine whether security requirements are appropriate, but this is outside the scope of operational assurance. (See Chapter 8.) Less formal audits are often called *security reviews*.

Audits can be self-administered or independent (either internal or external).[74] Both types can provide excellent information about technical, procedural, managerial, or other aspects of security. The essential difference between a self-audit and an independent audit is objectivity. Reviews done by system management staff, often called self-audits/ assessments, have an inherent conflict of interest. The system management staff may have little incentive to say that the computer system was poorly designed or is sloppily operated. On the other hand, they may be motivated by a strong desire to improve the security of the system. In addition, they are knowledgeable about the system and may be able to find hidden problems.

> A person who performs an independent audit should be free from personal and external constraints which may impair their independence and should be organizationally independent.

The independent auditor, by contrast, should have no professional stake in the system. Independent audit may be performed by a professional audit staff in accordance with generally accepted auditing standards.

There are many methods and tools, some of which are described here, that can be used to audit a system. Several of them overlap.

9.4.1.1 Automated Tools

Even for small multiuser computer systems, it is a big job to manually review security features. Automated tools make it feasible to review even large computer systems for a variety of security flaws.

There are two types of automated tools: (1) active tools, which find vulnerabilities by trying to exploit them, and (2) passive tests, which only examine the system and infer the existence of problems from the state of the system.

Automated tools can be used to help find a variety of threats and vulnerabilities, such as improper access controls or access control configurations, weak passwords, lack of integrity of the system software, or not using all relevant software updates and patches. These tools are often very successful at finding vulnerabilities and are sometimes used by hackers to break into systems. Not taking advantage of these tools puts system administrators at a disadvantage. Many of the tools

[74] An example of an internal auditor in the federal government is the Inspector General. The General Accounting Office can perform the role of external auditor in the federal government. In the private sector, the corporate audit staff serves the role of internal auditor, while a public accounting firm would be an external auditor.

are simple to use; however, some programs (such as access-control auditing tools for large mainframe systems) require specialized skill to use and interpret.

9.4.1.2 Internal Controls Audit

The General Accounting Office provides standards and guidance for internal controls audits of federal agencies.

An auditor can review controls in place and determine whether they are effective. The auditor will often analyze both computer and noncomputer-based controls. Techniques used include inquiry, observation, and testing (of both the controls themselves and the data). The audit can also detect illegal acts, errors, irregularities, or a lack of compliance with laws and regulations. Security checklists and penetration testing, discussed below, may be used.

9.4.1.3 Security Checklists

Warning: Security Checklists that are passed (e.g., with a B+ or better score) are often used mistakenly as proof (instead of an indication) that security is sufficient. Also, managers of systems which "fail" a checklist often focus too much attention on "getting the points," rather than whether the security measures makes sense in the particular environment and are correctly implemented.

Within the government, the computer security plan provides a checklist against which the system can be audited. This plan, discussed in Chapter 8, outlines the major security considerations for a system, including management, operational, and technical issues. One advantage of using a computer security plan is that it reflects the unique security environment of the system, rather than a generic list of controls. Other checklists can be developed, which include national or organizational security policies and practices (often referred to as *baselines*). Lists of "generally accepted security practices" (GSSPs) can also be used. Care needs to be taken so that deviations from the list are not automatically considered wrong, since they may be appropriate for the system's particular environment or technical constraints.

Checklists can also be used to verify that changes to the system have been reviewed from a security point of view. A common audit examines the system's configuration to see if major changes (such as connecting to the Internet) have occurred that have not yet been analyzed from a security point of view.

9.4.1.4 Penetration Testing

Penetration testing can use many methods to attempt a system break-in. In addition to using active automated tools as described above, penetration testing can be done "manually." The most useful type of penetration testing is to use methods that might really be used against the system. For hosts on the Internet, this would certainly include automated tools. For many systems, lax

procedures or a lack of internal controls on applications are common vulnerabilities that penetration testing can target. Another method is "social engineering," which involves getting users or administrators to divulge information about systems, including their passwords.[75]

9.4.2 Monitoring Methods and Tools

Security monitoring is an ongoing activity that looks for vulnerabilities and security problems. Many of the methods are similar to those used for audits, but are done more regularly or, for some automated tools, in real time.

9.4.2.1 Review of System Logs

As discussed in Chapter 8, a periodic review of system-generated logs can detect security problems, including attempts to exceed access authority or gain system access during unusual hours.

9.4.2.2 Automated Tools

Several types of automated tools monitor a system for security problems. Some examples follow:

- *Virus scanners* are a popular means of checking for virus infections. These programs test for the presence of viruses in executable program files.

- *Checksumming* presumes that program files should not change between updates. They work by generating a mathematical value based on the contents of a particular file. When the integrity of the file is to be verified, the checksum is generated on the current file and compared with the previously generated value. If the two values are equal, the integrity of the file is verified. Program checksumming can detect viruses, Trojan horses, accidental changes to files caused by hardware failures, and other changes to files. However, they may be subject to covert replacement by a system intruder. Digital signatures can also be used.

- *Password crackers* check passwords against a dictionary (either a "regular" dictionary or a specialized one with easy-to-guess passwords) and also check if passwords are common permutations of the user ID. Examples of special dictionary entries could be the names of regional sports teams and stars; common permutations could be the user ID spelled backwards.

---

[75] While penetration testing is a very powerful technique, it should preferably be conducted with the knowledge and consent of system management. Unknown penetration attempts can cause a lot of stress among operations personnel, and may create unnecessary disturbances.

- *Integrity verification programs* can be used by such applications to look for evidence of data tampering, errors, and omissions. Techniques include consistency and reasonableness checks and validation during data entry and processing. These techniques can check data elements, as input or as processed, against expected values or ranges of values; analyze transactions for proper flow, sequencing, and authorization; or examine data elements for expected relationships. These programs comprise a very important set of processes because they can be used to convince people that, if they do what they should not do, accidentally or intentionally, they will be caught. Many of these programs rely upon logging of individual user activities.

- *Intrusion detectors* analyze the system audit trail, especially log-ons, connections, operating system calls, and various command parameters, for activity that could represent unauthorized activity. Intrusion detection is covered in Chapters 12 and 18.

- *System performance monitoring* analyzes system performance logs in real time to look for availability problems, including active attacks (such as the 1988 Internet worm) and system and network slowdowns and crashes.

9.4.2.3 Configuration Management

From a security point of view, configuration management provides assurance that the system in operation is the correct version (configuration) of the system and that any changes to be made are reviewed for security implications. Configuration management can be used to help ensure that changes take place in an identifiable and controlled environment and that they do not unintentionally harm any of the system's properties, including its security. Some organizations, particularly those with very large systems (such as the federal government), use a configuration control board for configuration management. When such a board exists, it is helpful to have a computer security expert participate. In any case, it is useful to have computer security officers participate in system management decision making.

Changes to the system can have security implications because they may introduce or remove vulnerabilities and because significant changes may require updating the contingency plan, risk analysis, or accreditation.

9.4.2.4 Trade Literature/Publications/Electronic News

In addition to monitoring the system, it is useful to monitor external sources for information. Such sources as trade literature, both printed and electronic, have information about security vulnerabilities, patches, and other areas that impact security. The Forum of Incident Response Teams (FIRST) has an electronic mailing list that receives information on threats, vulnerabilities,

and patches.[76]

9.5 Interdependencies

Assurance is an issue for every control and safeguard discussed in this handbook. Are user ID and access privileges kept up to date? Has the contingency plan been tested? Can the audit trail be tampered with? One important point to be reemphasized here is that assurance is not only for technical controls, but for operational controls as well. Although the chapter focused on information systems assurance, it is also important to have assurance that management controls are working well. Is the security program effective? Are policies understood and followed? As noted in the introduction to this chapter, the need for assurance is more widespread than people often realize.

Life Cycle. Assurance is closely linked to the planning for security in the system life cycle. Systems can be designed to facilitate various kinds of testing against specified security requirements. By planning for such testing early in the process, costs can be reduced; in some cases, without proper planning, some kinds of assurance cannot be otherwise obtained.

9.6 Cost Considerations

There are many methods of obtaining assurance that security features work as anticipated. Since assurance methods tend to be qualitative rather than quantitative, they will need to be evaluated. Assurance can also be quite expensive, especially if extensive testing is done. It is useful to evaluate the amount of assurance received for the cost to make a best-value decision. In general, personnel costs drive up the cost of assurance. Automated tools are generally limited to addressing specific problems, but they tend to be less expensive.

References

Borsook, P. "Seeking Security." *Byte*. 18(6), 1993. pp. 119-128.

Dykman, Charlene A. ed., and Charles K. Davis, asc. ed. *Control Objectives – Controls in an Information Systems Environment: Objectives, Guidelines, and Audit Procedures*. (fourth edition). Carol Stream, IL: The EDP Auditors Foundation, Inc., April 1992.

Farmer, Dan and Wietse Venema. "Improving the Security of Your Site by Breaking Into It." Available from FTP.WIN.TUE.NL. 1993.

[76]For information on FIRST, send e-mail to FIRST-SEC@FIRST.ORG.

Guttman, Barbara. *Computer Security Considerations in Federal Procurements: A Guide for Procurement Initiators, Contracting Officers, and Computer Security Officials*. Special Publication 800-4. Gaithersburg, MD: National Institute of Standards and Technology, March 1992.

Howe, D. "Information System Security Engineering: Cornerstone to the Future." *Proceedings of the 15th National Computer Security Conference*, Vol 1. (Baltimore, MD) Gaithersburg, MD: National Institute of Standards and Technology, 1992. pp. 244-251.

Levine, M. "Audit Serve Security Evaluation Criteria." *Audit Vision*. 2(2). 1992, pp. 29-40.

National Bureau of Standards. *Guideline for Computer Security Certification and Accreditation*. Federal Information Processing Standard Publication 102. September 1983.

National Bureau of Standards. *Guideline for Lifecycle Validation, Verification, and Testing of Computer Software*. Federal Information Processing Standard Publication 101. June 1983.

National Bureau of Standards. *Guideline for Software Verification and Validation Plans*. Federal Information Processing Standard Publication 132. November 1987.

Nuegent, W., J. Gilligan, L. Hoffman, and Z. Ruthberg. *Technology Assessment: Methods for Measuring the Level of Computer Security*. Special Publication 500-133. Gaithersburg, MD: National Bureau of Standards, 1985.

Peng, Wendy W., and Dolores R. Wallace. *Software Error Analysis*. Special Publication 500-209. Gaithersburg, MD: National Institute of Standards and Technology, 1993.

Peterson, P. "Infosecurity and Shrinking Media." *ISSA Access*. 5(2), 1992. pp. 19-22.

Pfleeger, C., S. Pfleeger, and M. Theofanos, "A Methodology for Penetration Testing." *Computers and Security*. 8(7), 1989. pp. 613-620.

Polk, W. Timothy, and Lawrence Bassham. *A Guide to the Selection of Anti-Virus Tools and Techniques*. Special Publication 800-5. Gaithersburg, MD: National Institute of Standards and Technology, December 1992.

Polk, W. Timothy. *Automated Tools for Testing Computer System Vulnerability*. Special Publication 800-6. Gaithersburg, MD: National Institute of Standards and Technology, December 1992.

President's Council on Integrity and Efficiency. *Review of General Controls in Federal Computer Systems*. Washington, DC: President's Council on Integrity and Efficiency, October 1988.

President's Council on Management Improvement and the President's Council on Integrity and Efficiency. *Model Framework for Management Control Over Automated Information System*. Washington, DC: President's Council on Management Improvement, January 1988.

Ruthberg, Zella G, Bonnie T. Fisher and John W. Lainhart IV. *System Development Auditor*. Oxford, England: Elsevier Advanced Technology, 1991.

Ruthburg, Zella, et al. *Guide to Auditing for Controls and Security: A System Development Life Cycle Approach*. Special Publication 500-153. Gaithersburg, MD: National Bureau of Standards, April 1988.

Strategic Defense Initiation Organization. *Trusted Software Methodology*. Vols. 1 and II. SDI-S-SD-91-000007. June 17, 1992.

Wallace, Dolores, and J.C. Cherniasvsky. *Guide to Software Acceptance.* Special Publication 500-180. Gaithersburg, MD: National Institute of Standards and Technology, April 1990.

Wallace, Dolores, and Roger Fugi. *Software Verification and Validation: Its Role in Computer Assurance and Its Relationship with Software Product Management Standards*. Special Publication 500-165. Gaithersburg, MD: National Institute of Standards and Technology, September 1989.

Wallace, Dolores R., Laura M. Ippolito, and D. Richard Kuhn. *High Integrity Software Standards and Guidelines*. Special Publication 500-204. Gaithersburg, MD: National Institute of Standards and Technology, 1992.

Wood, C., et al. *Computer Security: A Comprehensive Controls Checklist*. New York, NY: John Wiley & Sons, 1987.

III. OPERATIONAL CONTROLS