The Essence of Information Assurance and Its Implications for the Ada Community

Michael McEvilley
Decisive Analytics Corporation
1235 Jefferson Davis Hwy Suite 400, Arlington, VA 22202 USA
001.703.414.5002
michael.mcevilley@dac.us

ABSTRACT

There is a growing need to engineer and operate critical systems with some level of measurable trust in the functions that make these systems secure. The term Information Assurance (IA) is now widely used to refer to the general concept of system security and this establishment of trust, but IA has yet to receive a precise definition. This paper discusses the concepts and principles that are essential to achieving IA objectives as a means to provide insights to the safety-critical Ada community regarding opportunities to impact the development of trusted security solutions.

Keywords: Information Assurance, Security Engineering.

1. INTRODUCTION

The development, integration and verification of security functions into complex systems requires the application of disciplined engineering practices. However, the concepts of computer security and the application of those concepts to the engineering of secure systems are less than 20 years old. In comparison, the disciplines of systems engineering and the engineering of safety-critical systems through application of systems engineering principles are over 50 years old [6]. As the need to engineer security in diverse networked and distributed systems increases, the challenge for security researchers and practitioners is to establish a security engineering discipline that results in cost-effective secure solutions that reduce risk to acceptable levels.

It is the intent of this paper to meet two objectives:

1. To provide a basis for understanding the widely used but not precisely defined term *Information Assurance (IA),* and
2. To focus on the technology portion of the information assurance equation and to draw insights to the increasingly visible overlap between implementing *secure* technology-based solutions and implementing *safe* technology-based solutions.

These objectives will be met by presenting a historical account of the development of foundational computer security concepts and the process through which trust is established for security products. From this foundational base the concepts of Information Assurance will be presented. Finally, the issues that overlap the security and safety critical communities will be presented to illustrate where opportunities exist for the Ada community to positively impact secure systems development.

2. HISTORICAL PERSPECTIVE OF COMPUTER SECURITY

2.1 Establishment of Cornerstone Security Concepts

Computer security has traditionally been defined as providing and ensuring the properties of *confidentiality*, *integrity* and *availability*. This suffices as a working definition. However, if we are to design and implement functions to implement these properties there must be precise definitions of the core security concepts and principles. Prior to 1985 no such concepts and principles existed; computer security focused primarily on physical processes and controls to restrict access to the hardware upon which software executed and on protecting the medium that enabled communication between distributed components of computer systems.

Research into the computer technology-based aspects of computer security began in mid-1970. This research culminated with the establishment of foundational concepts that defined security in terms of the properties that must exist in the security functions implemented in combinations of software and hardware[5]. In 1985, the US Department of Defense (DoD) published the Trusted Computer Security Evaluation Criteria (TCSEC) for applicability to computer systems utilized by the DoD [7]. These criteria, known as the "Orange Book", established three key things:

1. A conceptual basis and definition of the technical properties that constituted the foundation for trust in computer systems,
2. A hierarchical statement of the mandatory properties that an implementation must possess to be regarded as trusted based upon predefined operational policies and environments particular to DoD systems, and,
3. The basis for definition of a formal product evaluation process that provides a means for measuring the validity of claims of trust[1]

These base criteria were drafted for application specifically to operating systems and were later interpreted for application in the context of database management systems and networks. These

[1] Formal evaluation is a methodical design, implementation and test analysis campaign that verifies the correctness of the security properties of an implementation. The output of the evaluation is evidence that provides some level of assurance (i.e., confidence) that the implementation does in fact meet its requirements.

criteria were also crafted in response to the military security problem as defined by policy regarding the separation and isolation of information associated with multiple levels of sensitivity (e.g., Unclassified, Confidential, Secret, Top Secret). Subsequent to the Orange Book, evaluation criteria were developed in European countries and in Canada. The Orange Book and its various international derivatives have since been replaced by the Common Criteria for Information Technology Security Evaluation (ISO/IEC 15408)[2] which continues to serve as the basis for an international product security evaluation program [3].

2.2 Product Security Evaluation – Significant Gains/Significant Concerns

The product security evaluation focus is on verifying the correct implementation of the security functions independent of any specific operational context. This evaluation is conducted against a single instance of a product and the results of the evaluation serve as statement that the product meets its security requirements. The product security evaluation activity is viewed as a "clean-room/laboratory" verification activity because it excludes the operational environment in which the product might be employed[1]. The product security evaluation utilizes various types of evidence to substantiate the verdict that the specified level of trust has been established. This evidence includes requirements, design and test documentation, as well as, various user and administrative guidance documentation. Finally, product security evaluations are conducted so as to achieve some defined level of assurance. The variance in the assurance achieved comes through manipulation of three variables: *scope* – how much of the implementation is subject to evaluation, *depth* – to what level of detail is evidence developed and scrutinized to support the evaluation, and *rigor* – to what degree of formality is the evaluation evidence generated and scrutinized.

There are several concerns associated with the context of the product evaluation and the meaning of the results of a completed evaluation. These concerns include:

1. The product evaluation focuses on security for discrete products as opposed to the integration of products into a system.
2. The product evaluation excludes the contribution to security made by the physical installation environment and the processes governing use and operation of the product.
3. The product evaluation does not evaluate the actual configuration of the product as it would be configured to enforce a specified security policy in an identified operational environment.
4. The product evaluation is conducted once in a configuration defined by the evaluation team and the results of the evaluation were accepted as meaningful only in that specific configuration.
5. The product evaluation is a pass/fail activity – there is no concept of a conditional pass based upon the combined level of established trust and associated level of residual risk.

In response to these concerns separate and independent approaches to system security verification are employed to address the needs for establishing trust in the broader context of systems and for assessing the residual risk associated with the trust that is established. In the U.S., this activity is referred to as certification and accreditation. The certification portion establishes the basis for an argument that the system meets a set of security requirements in a manner that reduces residual risk to an acceptable level. The accreditation portion is the authorization given to employ the system for operational use.

The essential point from the above is that the concepts, criteria and processes that govern product evaluation do not sufficiently answer the critical questions regarding trust in networked and distributed systems. Recognizing that the Orange Book criteria were established in 1985, some 15 years have passed and the security community still struggles with the concept of system evaluation: how it is defined, how it is conducted, and what conclusions may be reached regarding assurance once the evaluation completes.

The new Common Criteria helps in many respects, as its base concepts and the generic nature of the criteria supports its application in a systems context. However, the primary focus for Common Criteria-based security evaluation remains to be that of product evaluation, and as such there is room for improvement to allow the CC to fully support a system specification and supporting evaluation process[2].

2.3 Concepts of Trusted Security Functions

Security evaluation focuses on the low-level functions that implement the stated security requirements. The term Trusted Computing Base (TCB) refers to the set of security functions that are the subject of the evaluation activity[3]. The TCB was defined as the totality of protection mechanisms within a computer system – including hardware, firmware and software – the combination of which is responsible for enforcing a security policy[7]. The TCB had to stand up to critical analysis and testing to ensure that it was an accurate implementation of its requirements. Those requirements reflected some or all of the following principles and concepts[4]

- **Security Policy** –the set of rules that define the interactions between subjects and objects and that defines how objects are managed, protected and distributed. The importance of the security policy is that every security function must support the enforcement of the security policy; otherwise that functionality constitutes an over-engineered solution. Conversely, should the security policy have aspects that are not enforced by the set of security functions then the security solution is under-engineered.
- **Identification & Authentication (I&A)** – the functions that authenticate the claimed identity of a subject. I&A may be as simple as a user name and an associated password, or may be as complex as the use of biometrics technology or strong encryption technology. A concept closely tied to I&A is *user-subject* binding. User-subject binding is the translation and

[2] While the CC may be used to specify the requirements for systems and used to support various system verification processes, there is no accepted methodology for conducting security evaluations of systems. As such, the reference to the CC as a basis for conducting product evaluations intentionally omits its use in the systems context. There are ongoing efforts to evolve the CC for applicability in system evaluation.

[3] The Common Criteria maintains the concept of the TCB, however the term used is Target of Evaluation (TOE) Security Functions (TSF).

[4] The terms subject and object are used to define the security concepts. Subjects are the active entity that causes information to flow or that change system state. Objects are the passive entity that contains or receives information.

binding of the attributes associated with an individual to the subject (i.e., process) that acts on behalf of that individual.

- **Access Mediation** – the functions that enforce a policy governing subjects, objects and operations. The policy will define the operations that a subject may perform on an object based upon a set of conditions represented by attributes associated with the subject and the object. The widely known access control policies are Discretionary Access Controls (DAC), Mandatory Access Controls (MAC) and Role-Based Access Controls (RBAC).
- **Audit** – the recording of the security relevant activity sufficient to support reconstruction of events and event sequences, to include the process/individual that initiated the event, times associated with the event and the outcome of the event. It should be noted that the concept of auditing and the association or analysis of information associated with the audit events has a cornerstone role in development of intrusion detection and response capabilities.
- **Object Reuse** – preventing a subject from gaining access to residual information remaining from an authorized operation performed by some other subject.
- **Non-Bypassability** – ensuring that the trusted functions can not be circumvented, that is, no action that is mediated by the trusted functions is allowed unless the trusted functions are invoked and return a verdict explicitly allowing that action.
- **Protection Domains** – ensuring that the trusted functions can not be tampered with. At a minimum, there must be a distinct separation between the trusted and untrusted execution domains. Every interface to the trusted domain must be known and its operation verified for correct behavior.
- **Minimization of Size & Complexity** – the trusted functions must be of limited size and must be designed and implemented in a manner that enables reaching a conclusive verdict as to their correct implementation.
- **Reference Monitor** – the reference monitor is an abstract concept comprised of the non-bypassability, protection domains and minimization of size & complexity concepts.

2.4 The Security Case – the Basis for Substantiating Trust

In general, we can view trust as the measure of confidence, or *assurance* that can be placed on the predictable occurrence of an anticipated event, or an expected outcome of a process or activity [8]. There are many variables that combine to establish a specific *contextual* meaning of trust. Reaching consensus on a contextual meaning of trust tends to be the major problem that system stakeholders deal with when developing and verifying security functions. Without a consensus definition of trust that is applied consistently across some set of development, verification and operations processes, the system stakeholders find themselves building incompatible component solutions. The incompatibility is often not discovered until the system is well into its integration, or worse, in operation.

The security case serves to establish a basis for trust where claims about the security properties of the system are substantiated through arguments based upon evidence that is generated or collected about the system. The security case may be generalized as containing the following:

1. Statement of the security problem that must be solved,
2. Statement of the solution to the security problem,
3. Substantiation that the components of the solution are both necessary and sufficient to solve the stated problem

The basis of assurance for the technology-based portion of a system lies in the above concepts and principles. The technology-based portion of the system must be seamlessly integrated with the non-technology portion to form a comprehensive and effective security solution – a solution that is substantiated by the security case. Information assurance deals directly with the issues in achieving and maintaining a comprehensive security solution over the life-cycle of the system.

3. INFORMATION ASSURANCE

The term Information Assurance, while widely used, has yet to be formally defined. While there is general consensus regarding the underlying activities that establish scope and context for obtaining information assurance, these activities continue to be conducted independently, with a narrow focus, and in the absence of information interchange[4]. There is increasing attention given to the nature and scope of IA. The IEEE Computer Society Task Force on Information Assurance and ACM SIG on Security, Audit and Control will co-sponsor the first IEEE International Information Assurance Workshop in the spring of 2003. A key goal of the workshop includes reviewing the definition of IA and the promotion of IA technology and corresponding standards.

So what is IA? It is perhaps easier to characterize IA in terms of the objectives that IA strives to meet than to precisely state a definition for IA. One such statement of IA as used by the DoD makes the following claim:

> The objective of IA is to provide the operations that protects and defends information, information-based processes, and information systems by ensuring their availability, integrity, authentication, confidentiality, and non-repudiation. This includes providing for the restoration of information systems by incorporating protection, detection, and response capabilities.

There are two interesting aspects of this statement: 1) the similarity in this statement and the definition of security presented in Section 2, and 2) the statement could be interpreted as focusing on the physical operations that establish and maintain the environment within which security technology solutions are employed. However, in recognizing the significant contributions to the concept of computer security made by the product evaluation community, we may conclude that there must be an element of *trust* established in the technology solutions that support IA objectives. It is therefore necessary to include the elements of trust resulting from years of product evaluation experiences.

Effectively, IA involves all of the *people, activities and technologies* employed to ensure that the fundamental properties of security (confidentiality, integrity, availability) are met throughout the life-cycle of the system. IA is conducted in a comprehensive “define, implement, verify” manner. IA has a larger scope than systems engineering as it extends beyond systems engineering to include the day-to-day operation and maintenance of the system.

Therefore, information assurance is not a process, methodology, or technology. It is an objective that is obtained through prudent application of appropriate processes, methodologies and technologies. It is subjective in nature and therefore, must be precisely defined to ensure that statements of IA objectives have meaning to those impacted by the processes associated with the development, operation and use of the system.

There issues addressed by information assurance are presented in terms of the activities of Assessment & Verification, Planning, Engineering, and Operations & Maintenance.

Assessment & Verification– In the broad sense, assessment and verification is where the security problem is defined and quantified and the security solution is verified. In the limited sense, assessment and verification are each conducted at defined points throughout all the IA activities. They are conducted to ensure that each and every activity supports or provides a necessary and sufficient part of the solution.

The assessment portion involves data collection to define the problem. This includes vulnerability, risk and threat identification and assessment, site surveys, penetration testing and analysis, and auditing. The verification portion includes testing, formal evaluation, certification and accreditation, auditing, network penetration and analysis.

Planning – The planning activities include: 1) the establishment of a non-technical infrastructure, that is, the development of policy, standards and procedures to govern all phases of the system life-cycle, 2) the establishment of the system management infrastructure, development and execution of education, training and awareness programs, and 3) planning for continuity of operations. Planning activities also include development of the statement of technical requirements.

Engineering – This is where the solution to the security problem is refined and implemented. It includes the development of the technical architecture and infrastructure. Discrete components are selected or designed, developed, and integrated into the architecture and infrastructure to construct the technology portion of the solution.

Operations & Management – This is where the security solution is employed as an enabler to the mission or business case. The activities include but are not limited to maintaining the security posture of the operational system, incident detection and response, and continuity of operations.

4. IA and the Ada COMMUNITY

IA serves to establish and maintain a security posture that is appropriate as defined by the responsible stakeholders. The previous sections intentionally focused on the technology-based aspects of IA and that theme will continue in this section where the challenges in meeting IA objectives are discussed.

It has been established that the discipline of computer security is relatively immature while at the same time the security problems are getting increasingly complex. The increased complexity is due to incorporation of new technologies, the ubiquity of 24/7 networked connections, and the increasing dependence on computer systems to provide critical functionality. There is now an increased interest in security outside of the traditional military and intelligence operations domains. The spectrum now includes the home user with continuous broadband Internet access; government organizations and commercial enterprises that operate in widely distributed wired and wireless environments; specialized industrial control systems and other real-time management systems that constitute critical infrastructures or that provide critical services.

In his testimony before the U.S. House of Representatives, House Science Committee, Dr W.A. Wulf asserted the need for the security community to step back and reassess where we are, where we are going, and to determine how and in what manner we can get to the where we must be[9]. Amongst others, his testimony included the following issues which are relevant to the discussion in this paper:

- **Security Technical Knowledge Base** – There is an insufficient technical base on which to build truly secure systems and the critical knowledge is held by very few academics, researchers and practitioners.
- **Security Definition** – We must redefine security and our security problems in manners appropriate to the context of the problem. This will prevent us from solving the right problem in the wrong context and in the wrong manner. The working definition of security continues to maintain a DoD perspective aimed at achieving DoD objectives.
- **Proactive Security Engineering** – We must change the current mode of implementing security. We tend to operate in a reactionary mode by relying on best practices to patch security problems. This approach does nothing to solve the fundamental problems – security engineering must be a pro-active effort.
- **Requirements Engineering** – We must engineer security requirements just as we engineer requirements for other critical solutions. It is often the case that security vulnerabilities result from the correct behavior of the system. The security vulnerability is not a result of a flaw in the design or operation of the system, the vulnerability results from a flaw in the security specification.

The security community has a significant challenge in addressing the above issues and can respond to the challenge by drawing on the experiences of other critical system engineering communities. The Ada community, as a component of the larger safety-critical community, has realized success in addressing some of the issues involved in developing reliable high assurance systems. There are opportunities to apply solutions that are effective in a safety-critical context to solve problems in the security-critical context.

A significant difference between security and safety lies in the philosophy that drives how security and safety are viewed in the overall system context. It is often the case that safety requirements receive highest priority such that they constrain the manner in which a system function is implemented. In contrast, security requirements are typically implemented as add-ons, in a reactionary manner, and have little if any influence over the functional posture of the system. This is likely to change as security becomes more embedded in the definition of what constitutes successful completion of business or mission functional requirements.

The specific areas in which the Ada community may impact IA lies in the processes of software design, software architecture, implementation and testing, and in the compilers, run-time, development, and assessment and measuring tools that support the process of developing high assurance software. The importance of injecting assurance into the various phases of the development process is illustrated by the widespread adoption of safety-critical guidelines such as DO 178B – Software Considerations in Airborne Systems and Equipment Certification [DO-178B ref DO-178B] and security specification criteria such as ISO 15408/Common Criteria [ref]. There is significant overlap in those two domain-specific documents and the concepts upon which they are based.

Establishing and Substantiating the Assurance Case

The safety case establishes evidence that provides a convincing and valid argument that a system is adequately safe for a given application in a given environment. The security case may be defined using the same text with simple substitution of security references where there are safety references. Conceptually, the safety and security cases are refinements of a more general assurance case concept.

Outstanding Issue: As the definition of IA is formalized and the concepts of system evaluation are refined, there is need to establish security trust concepts for networked and distributed systems. There is also the need to extend the security assurance case to incorporate the arguments and evidence essential to substantiating a claim of trust in the systems context.

System Architecture

Protecting the trusted security components from tamper by untrusted components is fundamental in establishing a security architecture. At higher levels of assurance there might be reason to ensure that trusted components are also protected from tamper by other trusted components. The safety critical equivalent to this is establishing an architecture that contains faults such that the anomalous behavior of one component is unable to affect the behavior of another component. Both needs may be served by isolation and partitioning, that is, implementing isolated execution domains and allocating critical processes to these domains.

A related concept utilized in safety critical solutions but not so for security critical solutions is that of multiple version dissimilar software implementations, that is, ensuring that fault modes are not replicated by the design.

Outstanding Issue: How might the architectural techniques and designs employed to meet safety-critical objectives be utilized to meet security objectives? How might safety-critical n-versioning and other techniques that prevent single-mode failures be utilized to meet security objectives?

Development Tools/Techniques

Many security vulnerabilities can be traced back to the development and verification tools used or the configuration and manner in which those tools are used. An example is the absence of compile and run-time constraint checking leading to buffer overflow vulnerabilities that may be exploited.

Some safety-critical implementations are developed using languages with limited instruction-sets and compilers/runtime environments that prevent the occurrence of buffer overflow conditions.

Outstanding Issue: Where should security solutions make use of development tools and techniques that eliminate some of the "common mode" security vulnerabilities? How might security solutions be architected to enable limited use of higher-assurance techniques where appropriate?

Verification Tools/Techniques

The 1970's recurring design that utilized 2-digits to represent the calendar year resulted in the Y2K problems 20+ years after their implementation. While not directly viewed as a security issue, the implications of a Y2K-like design fault might have significant security implications depending on the behavior of the system upon occurrence of the fault.

Outstanding Issue: In what way might the security community benefit from application of techniques developed for verification of safety-critical code in the context of security requirements implementation and verification?

5. CONCLUSION

There is an increasing need to engineer effective and measurable security into systems that provide critical functions and services. Security is now viewed as an enabler for mission and business critical operations rather than as a constraint to these operations. However, the diversity and complexity of system types, purposes, technologies and operational environments dictates that security solutions be engineered from the ground up – and not as add-ons implemented in reaction to the latest identified vulnerability.

Responding to this challenge will be difficult and requires that security engineering be performed at all levels of the product or system life-cycle. In parallel with this, new tools, techniques and processes for development and verification of security solutions are required to ensure that there is consistency and repeatability in these processes.

Many of the properties that make a system secure are the same properties that make a system safe. The tools and techniques used to develop and verify safe systems also have application in developing and verifying secure systems. The lessons learned and experiences of the Ada community in development of high assurance safe systems presents an opportunity to positively impact the direction and future of development of trusted secure systems.

6. REFERENCES

[1] Amoroso, E.G., Fundamentals of Computer Security Technology, PTR Prentice Hall (1994)

[2] Common Criteria for Information Technology Security Evaluation, http://www.commoncriteria.org/cc/cc.html.

[3] Common Criteria Recognition Arrangement, http://www.commoncriteria.org/registry/mr.html

[4] First IEEE International Information Assurance Workshop, http://www.ieee-tfia.org/iwia2003/cfp.html

[5] Gasser, M., Building a Secure Computer System, Van Nostrand Reinhold (1988)

[6] Leveson, N.G., Safeware: System Safety and Computers, Addison Wesley (1995)

[7] Trusted Computer System Evaluation Criteria (TCSEC), US Dept of Defense (1985)

[8] Whitmore, J.J., A Method for Designing Secure Solutions, IBM Systems Journal, Vol 40, No 2, 2001

[9] Wulf, W.A., Cyber Security: Beyond the Maginot Line, http://www.house.gov/science/full/oct10/wulf.htm