Integrating Information Security Engineering with System Engineering with System Engineering Tools

M. Douglas Higginbotham
Booz•Allen & Hamilton
higginbotham_doug@bah.com

Joseph G. Maley
Vitech Corporation
jmaley@vtcorp.com

Albert J. Milheizler
Department of Defense
ajmilhe@missi.ncsc.mil

Bernard J. Suskie
Booz•Allen & Hamilton
suskie_bernard@bah.com

Abstract

Users of Automated Information Systems (AISs) are becoming increasingly aware of the inherent risks associated with placing sensitive information on a system. Users are beginning to demand an assessment of the quality of security services offered because they need to make informed decisions on accepting certain levels of risk associated with protecting information they place on a system. By integrating an Information System Security Engineering (ISSE) process[1] into system development or system enhancement activities, system developers can satisfy user concerns. An ISSE process will identify the quality of security services needed by users; help identify security mechanisms to satisfy user needs; lead to an effective security design; identify the quality of security services offered by the actual system; and develop the documentation necessary to effectively market the security services offered by a system. An effective and cost efficient method for managing and providing discipline for the ISSE process is for system developers to use an automated system engineering tool. Such a tool significantly enhances the system security engineering team's ability to satisfy user security needs throughout the system design process.

Preface

Traditionally, ISSE is often treated as a specialty engineering effort focused on applying information protection techniques to previously established system architectures and design specifications. In addition, ISSE is often applied without investigating the fundamental protection needs of the Enterprise[2]. However, an empirical truth associated with ISSE is that to achieve functional and cost effective, application oriented INFOSEC capabilities, ISSE must be an integral, yet parallel part of the overall system engineering process. Not integrating ISSE with system engineering leads to weak security implementations and results in system users accepting an unknown level of risk when using the system. This paper outlines a process to integrate system security engineering at the system level; identifies the advantages of using a system engineering tool to manage and provide discipline for the design process; and identifies key documentation elements needed to assess how well the system satisfies an Enterprise's fundamental security needs. ISSE and appropriate system engineering tools can be effectively applied to those systems being developed using top-down, middle-out, and bottom-up system engineering processes.

---

[1] **ISSE is defined as applied system engineering with an emphasis on Information Security (INFOSEC) [4,5]**

[2] **Enterprise is defined as "organization(s) or Community of Interest (COI) which need to share information"**

1. Introduction

Users are ultimately responsible for accepting the risk of placing sensitive information on an AIS. Therefore, users may be liable for compromising the content, integrity, availability, etc., of information they place on a system. One primary goal of an ISSE process is to provide system users with an assessment of the quality of security services offered by the system. A user can then make an informed decision to accept the risk of information or system damage caused by accidental or malicious system use. The major phases of an ISSE process are a) identify system users' fundamental information protection needs; b) select and implement appropriate information protection mechanisms; and c) evaluate system protection mechanism effectiveness [1]. ISSE is implemented consistent with the system's mission statement, schedule, and program cost [2].

An effective overall system engineering process requires the identification of "Enterprise Mission Needs" which serves as the basis for system development. Security needs at the Enterprise Mission level should be identified during the planning stages of any top-down, middle-out, or bottom-up system development. The ISSE process documents top-level security requirements in the Enterprise Security Policy. A well thought out Enterprise Security Policy will identify the umbrella system security requirements and will accommodate different and changing user application security needs.

2. System engineering tool

The ISSE process is amenable to the application of automated system engineering tools. Using a system engineering tool will result in higher quality systems designed and produced in less time than conventional approaches [3]. System engineering tools support requirement analysis, behavioral analysis, architectural design; verification; document generation, and requirements traceability. One class[3] of system engineering tools implements an unambiguous executable requirements specification language. This class of engineering tools allows the design team to generate system/segment/component specifications that have both static and dynamic consistency [6]. A system engineering tool can improve an organization's "time to market" capability. More complex system designs become manageable and the impacts of changing requirements are much easier to determine. The system design is captured in an "entity, relationship, attribute" database while report scripts allow the engineering team to develop "multiple views" of the system. These views are used to gain insight into the design and to support any reporting criteria. This same class of tools greatly facilitates the identification of residual security risks [6]. The level of residual security risks governs whether an enterprise risk acceptance manager will allow the system to operate while containing sensitive information.

The CORE® System Engineering tool is an example of a tool useful in analyzing security threads. This tool helps identify "security relevant threads" and this information is applied to the security architecture development and the technical documentation needed for risk assessment. Figure 1 shows the results of analyzing one sample system thread for security relevant inputs, outputs, triggers, and functions**.** The diagram in Figure 1 is an enhanced functional flow block diagram.

3. System threads in the ISSE process

During the development of the system's behavior, a series of system threads are identified. These threads characterize the system interaction with all external systems and identify the information items that cross the system boundary. From the set of system threads, the design team derives the system's integrated behavior. The identified behavior is also traced to the originating functional requirements.

The Enterprise Security Policy is developed during the initial phase of the ISSE process and serves as the foundation for all subsequent ISSE activities. The ISSE process requires the translation of the Enterprise Security Policy into a set of leaf-level security requirements that are enforced by the system or through its operating environment. Leaf-level security requirements must be both necessary and sufficient to ensure protection of the users' information, the system's control information, the system's components, and external interfaces. These leaf-level security requirements address the management, protection, and distribution of sensitive information and the resources used to that information.

Using these security requirements, the design team examines the system operational threads to determine where and when the security requirements apply to the threads. Using a system engineering tool has significant importance in identifying and documenting the system threads. A tool simplifies the evaluation process for determining the security relevancy of these threads.

---

[3] Of the many tools being used for system engineering, the authors identified three, which fulfill the criteria necessary to perform system and system security engineering. These are CORE® from Vitech Corporation, RDD-100® from Ascent Logic Corporation and DCDS from the Army's Ballistic Missile Defense Center.

4. ISSE process

Integrating system security engineering at the system level using an ISSE process provides for the development of key documentation elements required to assess how well the system satisfies the Enterprise's security needs. The system design and the security design are not independent processes but parallel, compatible, and dependent processes. Both activities span the complete system life cycle. INFOSEC engineering is not subordinate to system engineering but an indispensable part of it. The nature of this INFOSEC engineering process is concurrent engineering performed layer-upon-layer with each additional layer refining the previous layer [6].

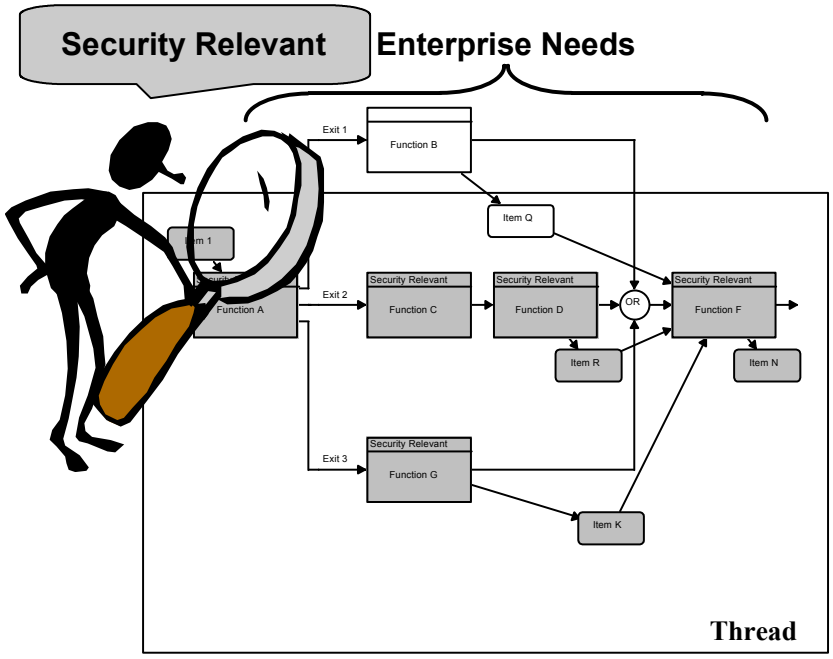

Figure 1 – Identifying Security Relevant Inputs, Outputs, Triggers, and Functions

As part of the layer-upon-layer refinement, the INFOSEC engineering process simultaneously develops the information necessary to evaluate the protection provided by the system. This information consists of a high-level operational concept, an architecture, and a description of how the system's security mechanisms satisfy the system's security requirements, see Figure 2. Although the process described in this paper can be performed without specialty tools, it is a distinct advantage to use a system engineering tool. A system engineering tool not only magnifies and leverages the capabilities of the design team, it also permits the production of the supervisory documents as a by-product. The system INFOSEC engineering process will yield the following technical documents:

- INFOSEC Operations Concept
- INFOSEC Architecture
- Theory of INFOSEC Compliance

The INFOSEC engineering process provides the security design information needed to prepare the three documents listed. The three documents are closely interrelated and are derived from the Security Policy, Figure 3**.** The INFOSEC Operations Concept document provides basic information used to prepare the INFOSEC Architecture and Theory of INFOSEC Compliance documents. The analysis of the INFOSEC Architecture, using the security requirements contained in the Enterprise Security Policy, comprises the bulk of the Theory of INFOSEC Compliance document. The following sections outline the process and provide guidance on developing the enterprise's INFOSEC documentation. The information presented is an adaptation of *Information Systems Security Handbook, Chapter 3*, using the consistent and complete approach expressed in [6].

4.1. System information

Describe the enterprise system**.** The initial phase of the INFOSEC engineering process is to develop the system description. This effort is accomplished through identifying and describing the system boundary and the relationship of the system to any external systems.

4.1.1. Identify and describe the enterprise system boundary. The description should provide an understanding of what functional elements lie within, what functional elements lie outside, and what items are transported across the boundary. When the functional behavior is allocated to components then the interfaces are automatically identified.

4.1.2. Describe the principal external systems and the information (items) that flow across the system boundary. The description should provide the design team's understanding of the external systems connecting to the system and the type and importance of the information being transported. As an example, the external systems may provide data, imagery, video, or audio files. Some information could contain sensitive information requiring protection by the system.

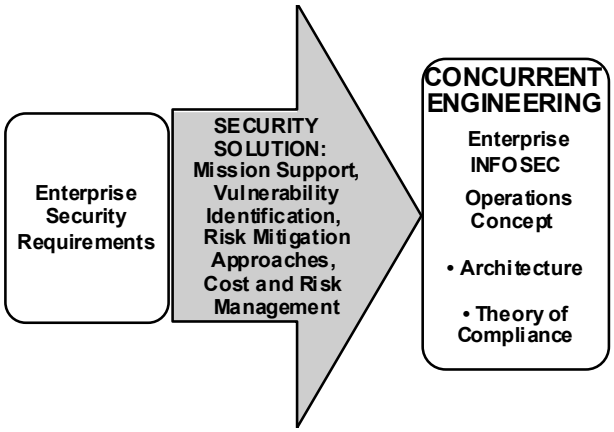

Figure 2 – INFOSEC Design Information

4.1.3. Identify and describe the security relevant information crossing the system boundary. Using the leaf-level requirements from the decomposed Enterprise Security Policy and the system boundary definition, the design team identifies for each external component the associated items (produced or consumed) and their INFOSEC relevancy. These input and output items are grouped into classes, where a class represents a collection of items having common characteristics.

4.2. INFOSEC operations concept

The INFOSEC Operations Concept bridges the enterprise security policy and the INFOSEC Architecture. The operational concepts depicted should be described in easily discernible terms comprehensible to users and accreditation authorities. In addition, the INFOSEC Operations Concept enables the security assessment team to determine the time sequencing and functional behavior of INFOSEC relevant activities. These activities also include events such as: the transport of sensitive information; establishment of system security parameters; the achievement of a secure state from a cold start, warm start, and after a fault; and the separation of security roles or duties.

The INFOSEC Operations Concept identifies what security is required at the lowest system design level. The INFOSEC Operations Concept is the cumulative functional behavior of each INFOSEC relevant system thread. A system thread describes, in stimulus-response terms, the system's behavior for a simplified system activity involving interactions with external systems. System behavior represents the time sequence or the steps that transform one or more system inputs into one or more system outputs [7]. The integration of these system and system security threads, along with some other leaf-level activities, form the system model.

4.2.1. Identify and describe the security relevant threads. Using baseline documentation and the defined system boundary, the INFOSEC design team begins to create operational threads that involve one or more INFOSEC relevant item classes and determine how the system should act on these items. Each input item is associated with one or more operational threads. An input thread examines how an INFOSEC relevant input item is acted on as it passes into and through the system. In some instances, the input item results in a system output. In other instances, the input item is consumed by the system. In addition, some output threads may exist that begin within the system and output INFOSEC relevant items to one or more external systems.

The set of INFOSEC relevant system threads is considered complete when all INFOSEC relevant input and output classes are identified and described. The external input and output classes contribute to defining the INFOSEC relevant information and control flows affecting the system. The INFOSEC Operations Concept document includes all INFOSEC relevant activities with each external system or element. For each INFOSEC relevant operational thread presented, all the information objects are identified and their security attributes described. Security attributes include, but are not limited to, sensitivity level, perishability, caveats, etc.

4.2.2. From the system threads determine which system threads and information flows are INFOSEC relevant (information and functions). INFOSEC relevancy is determined by prior knowledge, experience, or risk determination. In addition, each INFOSEC relevant thread is examined to determine what items and functions are security relevant within the thread.

4.2.3. Decompose the security relevant threads into a more detailed functional representation (functions, inputs, outputs, and triggers). These threads may be decomposed, as needed to gain additional insight into the behavior or INFOSEC characteristics of the thread. The functions, inputs, outputs and triggers are identified during each step along with determining their INFOSEC relevancy. The security relevant information is traced through the decomposed security thread to identify all derivative security relevant information and functions. The design team should determine the following as the information flow is traced through the thread:

- If the input or triggering information is security relevant, then the function is security relevant.
- If the aggregation of input or triggering information is security relevant, then the function is security relevant.
- If the function is security relevant, then the output may be security relevant. For example, an access control function produces a security relevant output, that is an access permission. A regrading function produces an unclassified output; the output is no longer security relevant.
- If the aggregation of functions is security relevant, then some functions within the aggregated set are security relevant.

4.2.4. Identify and describe applicable security services that provide protection for security relevant information. The design team determines what security services should be applied to protect security relevant information in each thread. This results in a top-level INFOSEC Operational Concept. The activities are identified and described by each thread. These activities are the functions and items (e.g., inputs, outputs, and triggers) that represent the information flow and control structure associated with the thread.

4.2.5. Identify and describe the preliminary allocated functional behavior of the security mechanisms that implement all the security services. The design team should strive to identify alternative security mechanisms for each component. These alternatives offer an opportunity to evaluate and select the most effective security mechanism. The components and the security functional behavior allocated to those components are described along with providing an analysis of the security mechanisms for performance, risk, and cost.

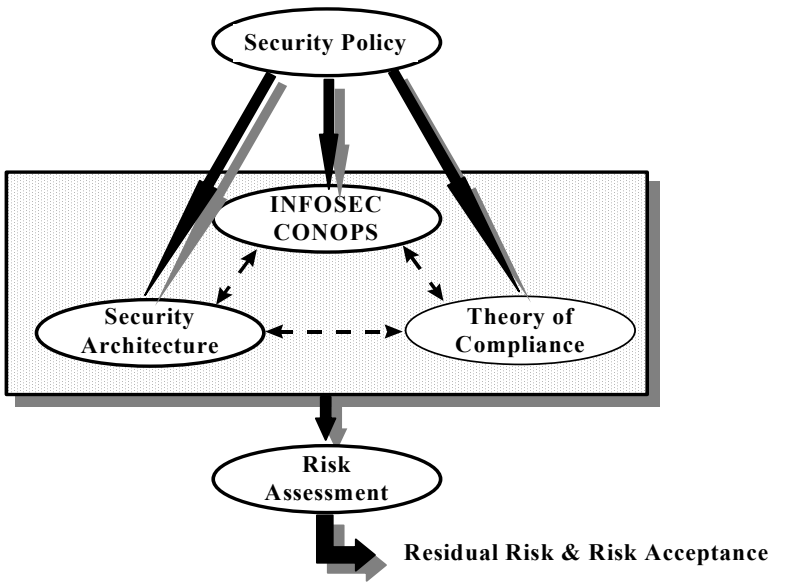

Figure 3 – System Security Engineering Documentation Relationships

4.3. INFOSEC architecture

The INFOSEC Architecture is the result of applying the INFOSEC engineering process and derives from the concepts captured in the INFOSEC Operations Concept and the system INFOSEC requirements. The INFOSEC Architecture shows the allocation of security behavior (the functions and control of services and mechanisms) to system components.

An INFOSEC Architecture is a system view that emphasizes information security. It provides insight into the allocated INFOSEC behavior and performance quality factors that satisfy both system and security requirements. It specifies where, within the context of the overall system architecture, INFOSEC behavior is found. It identifies the system configuration items, interfaces, and lower level components performing a security role and the INFOSEC behavior each supports.

The supporting INFOSEC architectural analyses identify interdependencies among INFOSEC-related components, behaviors, and external systems and then reconcile any conflicts among them. The INFOSEC Architecture enables the design team to evaluate viable architectural alternatives based on trade-off analyses among security, operational deployment, use, behavior, performance, and logistics issues. These system trades occur prior to selecting the best architecture.

Applying security services or security mechanisms to the INFOSEC relevant threads results in the incorporation of INFOSEC functionality into the threads. Further functional decomposition may be required before the system security design is complete. From the concurrent engineering perspective, the component architecture is developed in parallel and thread functions are tentatively allocated to high level components. Without having some functional allocation, a threat-vulnerability analysis is difficult to perform and weakens any risk assessment performed without it. These concurrent activities may be repeated at additional layers of decomposition.

To continue the INFOSEC engineering process and provide information for the theory of compliance, the following activities are necessary:

4.3.1. Assimilate all security relevant threads into a single integrated security relevant functional architecture. At some step in the thread analysis, the threads contain sufficient information to enable the engineering team to look for commonality and develop the integrated functional behavior of the system. INFOSEC functionality must also be simultaneously incorporated.

4.3.2. Integrate the single integrated security relevant functional architecture into the system functional architecture. The integrated functional behavior is allocated to components and the allocation is analyzed from an INFOSEC perspective. Adjustments to the functional behavior and the component architecture may occur to satisfy security as well as other system objectives. When the complete leaf-level functional behavior is determined, the system's functional architecture is achieved and is complete [8].

4.3.3. Allocate the functional security architecture onto the physical architecture. The INFOSEC functional behavior and its allocation to components must also be integrated with the overall functional behavior of the system and the system's physical components. This integration process also occurs concurrently with the INFOSEC engineering. The final integrated functional system behavior is the system INFOSEC functional architecture and the allocated functional behavior is the system INFOSEC architecture.

4.3.4. Identify and describe the final allocated functional behavior of the selected security mechanisms that implement all the system's security services. Indicate the components and the security functions allocated to those components and provide an analysis of security mechanism performance, risk, and cost. The system security architecture is again reviewed for its ability to satisfy the performance, risk, and cost objectives of the system.

4.4. Theory of compliance

The final phase of the INFOSEC engineering process prepares the Theory of INFOSEC Compliance document. This document describes how the allocated leaf-level INFOSEC requirements satisfy the system's INFOSEC requirements for the selected INFOSEC architecture. The Theory of INFOSEC Compliance describes how all the allocated INFOSEC functionality provides protection for the system.

Using the INFOSEC relevant threads, the design team illustrates how the item and function classes are specifically protected. The combination of the system's INFOSEC Architecture and the Theory of INFOSEC Compliance serve as security evaluation guides for certifying and accrediting the system. INFOSEC certification and accreditation testing and evaluation plans and procedures are derived from these documents. During implementation, any system changes must be checked for security impacts and update these documents whenever approved INFOSEC relevant changes occur.

The certification and accreditation team will use the Theory of INFOSEC Compliance document to determine

if the INFOSEC Architecture satisfies the security requirements. This document will form the basis of certification and accreditation testing.

4.4.1. Describe how the chosen security mechanisms protect security relevant system and user information from the security relevant system threads, identified in the security operations concept documents. The system's INFOSEC threads, as found in the INFOSEC Operations Concept document, may be used to illustrate how communications security services, security mechanisms, and control mechanisms are specifically provided. INFOSEC threads also help illustrate how system security vulnerabilities are mitigated.

4.4.2. Describe the residual risks to the system after the chosen security mechanisms are in-place. This description will provide the accreditors with valuable information on the capabilities of the system and how the system will react to different threat environments. The effort should involve a grading of the risk against a probability that the threat or attack will occur.

4.4.3. Describe at a high level, how the system will operate with the security mechanisms selected. This description should include operational and system user impacts imposed by the security mechanisms.

This description should walk the reader through the system functions related to INFOSEC and identify how the security works. In addition, the description should point out any impact to the system or user as the INFOSEC function is traced.

4.5. Risk assessment

Any security system, segment, or component should undergo an independent assessment of its security design and implementation. An assessment is necessary because no security design is impenetrable, some vulnerabilities will always remain. Therefore, some representative authority of the enterprise needs to evaluate the system's residual risk with respect to the system's operational mission. The enterprise user of cryptographic products and security services should use an independent system assessment team to determine the system's security suitability.

5. Summary

An INFOSEC System Engineering process has been presented that is fully integrated with an overall system engineering process. Not only does the process lead to a complete, consistent, and measurable security design; but, it is also repeatable and produces the material necessary to achieve an independent assessment of its security through a certification and accreditation process. It is an advantage to use an automated system engineering tool to leverage the engineering team and to develop more consistent and complete security designs.

6. Acknowledgments

This work was sponsored by the Department of Defense under contracts MDA904-96-C-1553 and MDA904-97-C-0312. The authors thank the anonymous reviewers whose comments improved this paper.

7. References

[1] *Information Systems Security Engineering Handbook*, National Security Agency, December 17, 1993, pp. 2-5 – 2-6

[2] *Information Systems Security Engineering Handbook*, National Security Agency, December 17, 1993, pp. 2-1

[3] Oliver, D., Kelliber, T., Keegan, J., Jr., E*ngineering Complex Systems with Models and Objects*, NY: McGraw-Hill, 1997, p. 2

[4] *Information Systems Security Engineering Handbook*, National Security Agency, December 17, 1993, pp. 2-6 – 2-15

[5] *Department of Defense Information Technology Security Certification and Accreditation Process, Enclosure 2*, November 1997

[6] J. E. Long, *CORE® Basic Class Training Material*, Vitech Corporation, 1995-1997

[7] Oliver, D., Kelliber, T., Keegan, J., Jr., E*ngineering Complex Systems with Models and Objects*, NY: McGraw-Hill, 1997, pp. 11, 133 – 134

[8] Oliver, D., Kelliber, T., Keegan, J., Jr., E*ngineering Complex Systems with Models and Objects*, NY: McGraw-Hill, 1997, pp. 137 – 14