Beyond the Preoccupation with Certification & Accreditation

Kevin Esser

Table of Contents 1
GSEC Opt 1 V1-4 Esser Final 10 Jan 05.doc 2

Beyond the Preoccupation with Certification & Accreditation

A Guide to Conducting Information Assurance Systems Engineering During the Development of Tactical Systems

GIAC Security Essentials
Certification (GSEC)
Practical Assignment
Version 1.4c
Option 1

Kevin J. Esser

10 January 2005

Table of Contents

Abstract……..3
1.0 Introduction……4
2.0 Scope and Methodology……..4
3.0 Concept Refinement Phase……6
4.0 Technology Development Phase……..12
5.0 System Development and Demonstration Phase……..16
6.0 Production and Deployment Phase……..18
7.0 Operations and Support Phase……19
8.0 Conclusion……20
References……..……21

Abstract

Seeking and achieving formal Certification and Accreditation of systems designed for use within the Department of Defense is a statutory requirement and a necessary part of a system's overall Information Assurance program. A singular focus on this "process" objective, however, too often overshadows critical Information Assurance engineering activities necessary during system design.

This problem is particularly acute in tactical system developments. This guide attempts to chart a course for the tactical system developer that interlaces crucial Information Assurance activities alongside standard Department of Defense acquisition and systems engineering design activities, ensuring security features are standard elements of system design and life-cycle supportability plans.

1.0 Introduction

Experience has shown that in tactical system developments, design and engineering evolutions focus heavily on meeting performance requirements and tailoring system characteristics to achieve peak operability. Tactical systems are engineered to meet unique customer design requirements that facilitate deployment and reliable operation in unusual environments. In striving to deliver systems that provide these customer-desired capabilities, Information Assurance (IA) elements are often overlooked or purposely pushed aside in favor of features that have tangible utility for the warfighter. As a result, compliance with IA guidelines is usually a low priority and often seen as a hindrance in meeting desired performance requirements.

In the past, when tactical systems were largely proprietary developments deployed within stove-pipe or circuit connectivity models, a reduced attention to IA tenets was deemed acceptable – in fact the term "Security by Obscurity" was often used to justify this position. Times have changed, however, and the focus now is clearly on fielding integrated, network-centric systems that connect military forces world-wide via the "Global Information Grid". To meet this goal, tactical systems have transitioned to software, hardware, network protocols and topologies developed primarily by and for a non-military commercial market base. These new realities plainly require that systems be engineered with a strong IA posture as a critical design objective.

Striking a balance between providing optimum performance and a robust IA posture is not an easy task, but can be made easier by following a disciplined IA systems engineering process and conducting critical security design activities at key points during system development.

This guidebook is intended as a reusable compendium that provides valuable IA engineering insight for tactical system acquisition professionals and developers at all stages of system design and development.

2.0 Scope and Methodology

There are many possible viewpoints from which to approach IA engineering and design activities – e.g. there are IA-focused product developments (e.g. Intrusion Detection/Prevention Systems and other security appliances), cryptographic device developments, standard client/server office environment system deployments, etc. I will focus on the integration of IA services into military tactical systems because they represent a unique subject where little guidance is available to direct the developer in IA implementation, from system conception through deployment.

For purposes of this guide, the term tactical system shall be synonymous with the definition of a "Mission Critical Information Technology (IT) System" as defined by Department of Defense (DoD) Instructions 5000.2 and 8500.2, i.e. a system "the loss of

which would cause the stoppage of warfighter operations or direct mission support of warfighter operations"[1] and where there exists "IT interconnections to the Global Information Grid."[2] In other words, an IT system crucial to warfighting with external data connectivity requiring focused attention to IA design and engineering concepts.

Each section is formatted in a similar fashion. The title headings are drawn from the acquisition phases defined by DoD Instruction 5000.2 and are then followed by a table that calls out a summary of the standard acquisition and systems engineering activities defined by chapter 4 of the Defense Acquisition Guidebook (DAG)[3]. The DAG was released and posted online in October 2004 "…as a companion to the revised acquisition policy documents, DoD Directive 5000.1 and DoD Instruction 5000.2…"[4]. The final column in this table calls out a number of the most critical IA design and engineering activities necessary at that stage in the tactical system's development. Each of those critical elements is then discussed in detail.

This format was chosen for several reasons: (1) to ensure IA activities are placed within a context that is widely understood across the DoD community, (2) because IA is becoming an increasingly more visible part of acquisition milestone decisions, and (3) to illustrate that IA design and engineering activities are most effective when carried out alongside other standard systems engineering evolutions.

This guidebook does not include an exhaustive discussion of the DoD acquisition process nor does it provide a listing of all potential activities called out by a comprehensive "Information Systems Security Engineering (ISSE)" process. Although recommendations regarding the application of ISSE models are made at several points, the focus here is on highlighting the most critical and oft-neglected IA activities.

Although it is rare that a user of this guide would be entering at the initial "concept refinement stage", the reader should be able to tailor the IA engineering activities called out here just as they would tailor other elements of the system's acquisition or engineering activities to their specific systems' development approach.

[1] United States. Department of Defense Instruction 5000.2, "Operation of the Defense Acquisition System." 12 May 03. 3 Jan 05. <http://akss.dau.mil/dag/DoD5000.asp?view=document&doc=2>

[2] United States. Department of Defense Instruction 8580.1, "Information Assurance in the Defense Acquisition System." 9 Jul 04. 3 Jan 05. <http://www.dtic.mil/whs/directives/corres/html/85801.htm>

[3] Defense Acquisition Guidebook Web Page. 2004. United States Department of Defense. 3 Jan 05. <http://akss.dau.mil/dag/DoD5000.asp?view=document>

[4] "DoD Publishes Defense Acquisition Guidebook," DefenseLINK News Release No. 1025-04. 14 Oct 2004. 3 Jan 05. <http://www.defenselink.mil/releases/2004/nr20041014-1389.html>

3.0 Concept Refinement Phase

DAG Defined Acquisition / Systems Engineering Activities	Critical Information Assurance Design and Engineering Activities
• Key Inputs: ○ Initial Capabilities Document ○ Analysis of Alternatives • Analyze Operational Capabilities and Environmental Constraints • Develop Performance and Functional Objectives • Identify Enabling and Critical Technologies • Key Outputs ○ Preliminary System Specification ○ Test & Evaluation Strategy ○ Systems Engineering Plan ○ Analysis of Alternatives	1. Conduct system-specific Threat Assessment 2. Conduct IA focused requirements discovery, analysis and allocation 3. Conduct Certification & Accreditation path assessment 4. Integrate Information Assurance requirements and Certification & Accreditation path assessments into mainstream acquisition planning

3.1 System-Specific Threat Assessment

IA focused threat assessments for tactical systems are often conducted by third party organizations that do not possess a detailed understanding of the system's architecture, environment or employment. As a result, baseline threat reports are usually too generic to be a useful tool for the security engineer. The system-specific threat assessment should specifically identify and prioritize the threats most relevant for that system. Without a detailed and quantified understanding of the threats and attack types judged to be the most credible, the security engineer will be unable to utilize the threat assessment as a tool in his decision-making toolbox while conducting system design trade-off analyses. The risk, aptly described in a recent Carnegie Mellon CyLab research report, is that the proposed IA infrastructure (functionality, components and overall architecture) will suffer from "…either overkill – where solutions suggested are stronger, less efficient, and more costly than needed – or under kill – where solutions do not adequately address the mission-relevant threats."[5]

One can generate an almost infinite list of threat agents (e.g. script kiddies, foreign militaries, professional hackers, terrorist networks, malicious insiders), attack methods (e.g. denial of service attacks, physical attacks, virus/worm planting) and motives (e.g. data compromise or modification, system reconfiguration or degradation, obtaining

[5] Moore, Andrew P. "Analyzing the Threat Dynamics of Complex Networked Systems." Carnegie Mellon 2004 CyLab Research Projects. 3 Jan 05. <http://www.cylab.cmu.edu/default.aspx?id=282>

enhanced privileges). What the tactical system threat assessment must do to meet its unique needs is evaluate relevant threat agents, their methods, their motives, and their potential to (1) cause the loss of system functionality – thereby endangering a system's mission, and (2) cause the compromise of data critical to the survival of the system or its attendant personnel. To do this and meet the objective of providing a useful tool to the security engineer requires quantitatively ranking the risk of each threat agent.

A notional methodology for doing such an assessment, inspired by the threat attributes used in Bradley J. Wood's Insider Threat Model,[6] might look like the following:

Risk Index · Threat Agent		Feasibility of Access		Knowledge of System		Skill Level		Risk Tolerance		System Impact
X	=	(1 Hard -5 Easy)	+	(1 Lo – 5 Hi)	+	(1 Lo – 5 Hi)	+	(1 Lo- 5 Hi)	+	(1 Lo- 5 Hi)

The numeric values can be of any range, I have used 1-5 here. Definitions of the variables in the model could also be tailored to fit the specific system in question – the definitions I have used for the following two examples are:

(a) *Feasibility of Access:* Relative numeric rating of the attack vectors available to the threat agent (e.g. via external Wide Area Network (WAN) vs. insider with terminal access)

(b) *Knowledge of System:* Relative numeric rating of threat agent's knowledge of a particular tactical system and its IA features (e.g. part time hacker conducting random port scans vs. member of system design team)

(c) *Skill Level:* Relative numeric rating of the penetration / infiltration skill level of the threat agent (e.g. script kiddy vs. trained penetration specialist)

(d) *Risk Tolerance:* Relative numeric level of risk the threat agent is willing to tolerate to achieve his objectives (e.g. teenager experimenting with a new tool vs. state-sponsored espionage agent with ideological motivations)

(e) *System Impact:* Relative numeric rating of the threat agent's perceived ability to impact system functionality or compromise data (e.g. direct impact to mission critical functionality or data vs. impact only to non-sensitive administrative data)

The two sample analyses below show how the model could be implemented and what conclusions can be drawn:

(a) *Threat Agent X:* Nation state-backed espionage agent with specialized network security training, only attack avenue to system is through a WAN Security Operations Center that operates an actively managed firewall, intrusion detection appliances, a load-leveling server configuration and virus detection software. Desires access to classified data necessary to operate the system.

[6] Wood, Bradley J. "An Insider Threat Model for Adversary Simulation." SRI International, Cyber Defense Research Center, System Design Laboratory Publications. 12 Jul 00. 3 Jan 05. <http://www.csl.sri.com/users/bjwood/Insider_threat_model_v02.pdf>

Risk Index · Threat Agent X (15)	=	Feasibility of Access (1)	+	Knowledge of System (1)	+	Skill Level (5)	+	Risk Tolerance (5)	+	System Impact (3)		

(b) *Threat Agent Y:* Knowledgeable insider working as an operator of the deployed system, has access to loosely controlled role-based administrator account, but only basic networking experience. Wishes to make an "unapproved" change to the system configuration to improve performance.

Risk Index · Threat Agent Y (17)	=	Feasibility of Access (5)	+	Knowledge of System (5)	+	Skill Level (1)	+	Risk Tolerance (1)	+	System Impact (5)		

Both threat agents pose legitimate threats to the system, one with malicious and one with non-malicious motives. But without using a model that attempts to quantify the magnitude of that threat, it may not be clear that although Threat Agent X is highly skilled and has a high risk tolerance, the knowledgeable insider (Threat Agent Y) with easy access and a higher potential impact has the higher overall risk index.

Once the comprehensive threat assessment is complete, the security engineer will have a tool that can be used to ensure system design choices maximize system performance while also building an IA infrastructure focused on defeating the most relevant IA threats.

3.2 IA Focused Requirements Discovery, Analysis and Allocation

In the DoD environment, the majority of the IA requirements for tactical and other military systems are called out in DoD Directive 8500.1, "Information Assurance"[7], and DoD Instruction 8500.2, "Information Assurance Implementation"[8]. But the IA requirements discovery task does not end there. Additional requirements may also be imposed due to cross-domain interconnections (i.e. interfaces between systems at different data classification levels), due to connectivity with global military WANs, or due to the classification of data processed within the system. And because tactical systems are deployed in high stress, high tempo environments, IA requirements discovery activities should also include a survey of user requirements. The end-user of a tactical system will likely have specific IA operability requirements based on the anticipated employment of the system. Examples might include requirements mandating the centralized storage of sensitive data, providing access to multiple

[7] United States. Department of Defense Directive 8500.1, "Information Assurance." 24 Oct 02. 3 Jan 05. <http://www.dtic.mil/whs/directives/corres/pdf/d85001_102402/d85001p.pdf>

[8] United States. Department of Defense Instruction 8500.2, "Information Assurance Implementation." 6 Feb 03. 3 Jan 05. <http://www.dtic.mil/whs/directives/corres/pdf/i85002_020603/i85002p.pdf>

classification levels of data at the same workstation, or employing user interfaces that make IA functions normally managed separately - accessible from one terminal.

In addition, government regulations now require that IA-enabled components and cryptographic devices be chosen from approved lists of certified products. The Common Criteria Evaluation and Validation Scheme, administered by the National IA Partnership, was established to ensure that "…products acquired by U.S. Government Departments and Agencies be subject to a standardized evaluation process, which will provide some assurances that these products perform as advertised."[9]

Once evaluated, IA-enabled components are assigned an Evaluated Assurance Level that denotes the rigor with which the products were tested and then placed on a validated products list. IA-enabled components within the tactical system (e.g. operating systems, firewalls, routers, etc.) must be chosen from this list. Cryptographic devices must meet validation requirements established by the National Security Agency. If non-validated products are desired, schedule and resources must be made available to push them through one of these evaluation processes. In a design environment where performance and operability requirements have the highest priority, choosing components from the pool of Common Criteria validated products may force trade-off studies and/or additional research & development efforts best completed early in the design definition process.

Once the superset of IA requirements is finalized, it must be analyzed and placed within the context of the system. The security engineer must determine how tactical system design and environment constraints will impact IA requirements allocation. Typical tactical system constraints include limited equipment space and weight allowances, limited hotel services (power, cooling), aperiodic external connectivity via low bandwidth circuits, etc.

For example, how will the system comply with requirements for intrusion detection, firewall services and secure routing when the space available for these functions collectively is the size of a shoebox? How will the system meet a requirement for frequent anti-virus signature updates when access to the necessary connectivity is only available every 30 days? Issues like these must be addressed for the allocation of IA requirements to software, hardware, interfaces, training, procedures, etc. to be effective.

In summary, the assessment of IA requirements within the context of the unique tactical system environment must be a key part of design definition activities.

[9] United States. National Security Telecommunications and Information Systems Security Policy No. 11, Revised Fact Sheet, "National Information Assurance Acquisition Policy." Jul 03. 3 Jan 05. <http://niap.nist.gov/cc-scheme/nstissp_11_revised_factsheet.pdf>

3.3 Certification & Accreditation Path Assessment

Even though the impetus for this guidebook is to refocus the "IA task" in the minds of tactical system developers from getting through the statutory Certification & Accreditation (C&A) process wickets to carrying out disciplined IA design and engineering activities, achieving successful C&A is still a necessary requirement in the DoD domain.

C&A processes and documentation formats change frequently over time, are usually dictated by data classification and/or system type, and are often dominated by the personal preferences of the approving and certifying authorities. As a result, this discussion will not focus on a specific C&A process (e.g. the DoD IT Security C&A Process (DITSCAP), the DoD Intelligence Information System (DoDIIS) C&A Process or the new DoD IA C&A Process (DIACAP)), but will highlight the key process-agnostic activities essential for placing tactical systems on a successful path to achieving C&A.

There are usually many organizations with influence on the C&A process for tactical systems, each of which has its own objectives and standards. A successful C&A effort requires the cooperation of all parties, so it is essential that the security engineer begin the C&A assessment by gaining an understanding of the viewpoints of the various parties and when they may act as help or hindrance on the path to C&A.

The following roles and their associated responsibilities and motivations are common:

(a) *Program Manager:* Responsible for overall program execution, compliance with all requirements (not just IA requirements), cost control, and meeting schedule milestones. The program manager's primary objective is delivery of the system. He/She will dial back functional implementations when cost or schedule constraints require it. IA implementation is often at the top of the "hit" list when performance requirements require additional resources to complete.

(b) *End Users:* Responsible for employing the system once delivered. The end user is primarily concerned with system operability, mission suitability and long-term supportability. The end user's primary objective is to influence the design in ways that deal with safety, reliability and ease of use. The end user may see successful certification of the system as in their best interest, but will only tentatively give up tangible needs for this more abstract objective.

(c) *Resource Sponsor:* Responsible for allocating the funds and other resources necessary for the program manager to develop the system. The resource sponsor is concerned most with the cost/benefit ratios of design decisions. As the final word in prioritizing and validating requirements and capabilities, if the resource sponsor does not have a strong IA advocate lobbying for a robust IA engineering effort, other elements will likely be given higher priority.

(d) *Certification Authority:* Usually an organization that is responsible for reviewing the IA implementation of systems across an entire military service. They are responsible for making a technical recommendation that a system is compliant with

applicable IA requirements. The certification authority is the IA standards bearer and must judge each system against a relatively rigid set of principles. As such, they are often hard-pressed to make concessions regarding IA requirements compliance for unique tactical operational or environmental reasons.

(e) *Designated Approving / Accrediting Authority:* The Designated Approving / Accrediting Authority (DAA) is responsible for making the final decision approving the system for operation. The DAA's primary concern is the residual IA risk remaining within the system after all design work and testing is complete. As the one upon whose shoulders rests the most responsibility, most DAAs want the highest priority IA vulnerabilities and risks to be mitigated prior to deployment.

Because their interests are often in competition, it is essential that the system security engineer get these organizations together early and often during the design process to ensure each develops an understanding of the tradeoffs that go into system design decisions, and is able to voice any concerns regarding IA implementations. Without their buy-in of the IA approach early, gathering the support necessary to get the necessary C&A approvals will be difficult.

It is then necessary to ensure that all the requisite C&A processes are identified. A tactical system that processes multiple classifications of data may have to weave its way through multiple C&A processes and satisfy multiple DAAs. Cross-Domain implementations or multi-level security architectures require adherence to yet more specialized C&A processes. It is essential that all C&A exposures are understood up front so that all respective process requirements can be properly scoped.

One of the most effective ways of ensuring harmony amongst each of these organizations and across all of the necessary processes is to establish dedicated certification authority and/or DAA liaisons within the cognizant system program office. These liaisons are able to advise on a near real-time basis whether or not the certifier or accreditor will accept a proposed configuration or recommend an alternative. By including such liaisons as part of the design process from beginning to end, IA-relevant design decisions will be made in a cooperative fashion making the achievement of C&A much more likely.

Regardless of the specific C&A process required, it is important the security engineer properly allocate the resources required to steward the system through the required steps. IA documentation development, risk assessments, testing, etc. must be done in concert with other system development activities. Attempting to rush the process by compressing activities together as the system nears deployment is a recipe for failure.

3.4 Integrating IA Requirements and C&A Path Assessments Into Mainstream Acquisition Planning

Threat assessments, focused IA requirements analyses and C&A path assessments are of little use if the results are not seamlessly folded into the mainstream acquisition and systems engineering activities that drive tactical system design and development.

IA requirements must be integrated into system and component specifications alongside other performance requirements. Systems engineering management plans must integrate IA engineering activities and required C&A tasks where appropriate with standard systems engineering activities. Test & Evaluation plans need to trace in test events to demonstrate compliance with IA requirements just as they do for performance, operability, environmental and other requirements.

As software unit test cycles and component integration testing are mapped out, any IA-unique testing or C&A evaluations need to be identified in order to make use of the same test bed windows of opportunity and find/fix/repair cycles. And as periodic system configuration baselines are established, any IA-specific configuration elements (e.g. operating system (OS) service configurations) must be documented and carefully configuration managed as essential parts of the overall baseline.

4.0 Technology Development Phase

DAG Defined Acquisition / Systems Engineering Activities	Critical Information Assurance Design and Engineering Activities

• Key Inputs:
  ○ Initial Capabilities Document
  ○ Preferred System Concept
• Demonstrate Enabling / Critical Technologies
• Demonstrate System Functionality
• Demonstrate Integrated System Performance Relative to Performance Specifications
• Key Outputs:
  ○ System Performance Specification
  ○ Test & Evaluation Master Plan
  ○ Technology Readiness Assessment
	1. Develop secure software development strategy
	
	2. Integrate system and Information Assurance-focused Test & Evaluation
	
	3. Identify and codify recertification expectations

4.1 Secure Software Development Strategy

To truly embed security at the core of a system, its software development activities must incorporate secure software development techniques from the onset. Even a short discussion of secure software development methodologies could cover many volumes, but there are several common elements that must be addressed during each tactical system development.

(a) Ensure new software development is entrusted to organizations that follow recognized methodologies. Good software engineering is a prerequisite for the production of secure software code. A disciplined approach should include a progression from requirements analysis to preliminary and detailed design phases and proceed through coding, unit testing, component integration and testing and finally subsystem and system integration testing.[10] Mandating developer compliance with established software development process standards (e.g. Software Engineering Institute Capability Maturity Models, International Organization for Standardization 9000 family) is a must for complex systems.

(b) Require that software development take into account universal guidelines for secure software development, including: input/output validation, use and reuse of trusted components, separation of privileges, defense-in-depth, secure fail modes, encryption of as much internal communication as possible, etc.[11]

(c) Too often the traditional mindset that "devices" will handle the bulk of IA requirements compliance has dominated, but it is imperative that developers choose an OS that will not only provide the performance required, but will also lay the foundation for the system's overall IA requirements compliance.[12] OSs can vary significantly in the way they handle memory addressing, mediate access between subjects and objects, authenticate users, control application to kernel interactions, etc. To ensure compliance with DoD OS configuration standards, the security engineer must evaluate how the OS provides this security functionality in addition to how it satisfies desired performance requirements.

(d) The timely response to vulnerabilities as they are discovered has become one of the most vexing IA management tasks for those associated with tactical systems. As discussed above, security-conscious software development and OS selection will go a long way to reducing potential vulnerabilities, but any system (especially those using commercial elements that are popular hacker targets) will have to deal with patching.

Planning for vulnerability response and patch development must be a part of software-related planning during system design maturation. The following post-delivery capabilities must be addressed concurrent with system functional definition and demonstration:

- *Development and validation of vulnerability patches.* Lab facilities, personnel and funding must be available to support patch development and testing of DoD or service-required patches for every deployed system baseline over their entire life-cycles. Many aging systems today remain at risk due to well-known

[10] Pfleeger, Charles P. Security in Computing, 2nd Edition. Upper Saddle River: Prentice-Hall, 1997. Page 215.
[11] "Improving Security Across the Software Development Life Cycle." National Cyber Security Partnership Task Force Reports. 1 Apr 04. 3 Jan 05. <http://www.cyberpartnership.org/SDLCFULL.pdf>
[12] Harris, Shon. All-In-One CISSP Certification Exam Guide. Berkely: Mcgraw-Hill/Osborne, 2002. Page 704.

vulnerabilities because the lab facilities and resources needed are no longer available.

- *Patch delivery.* Tactical systems often present unique challenges for the delivery of any kind of software update. Those that are time-sensitive are even more difficult. Whatever the method chosen (transportable media, network connectivity, etc.) - limited patch receipt windows of opportunity, low bandwidth, and other conditions may require advanced planning and coordination.
- *Patch Implementation*. The end user must have the necessary system knowledge, experience and local roll-back capability to be able to reliably apply software patches. If any of these elements is missing, personnel may need to be sent to the tactical system's location from off-site to install important patches.
- *Verification and Reporting.* Whether it be for local purposes or to comply with DoD reporting requirements, accurate compliance reporting is essential for both retaining an accurate picture of the system's IA posture and maintaining strong software configuration management. Ensuring dependable processes are established for verifying and reporting patch installation compliance should not be overlooked.

Emphasizing secure software development and maintenance of a secure posture over the entire life cycle is important for any system, but is particularly vital for tactical systems that prize reliability and availability above many other attributes.

4.2 Integration of System and Information Assurance-focused Test & Evaluation

As system functional and technology demonstrations wind down and the full scope of the system development process becomes clearer, it is necessary to ensure requisite validation of IA functionality is conducted incrementally with other system validation activities. The initial step in this process is to ensure that the IA implementation defined to meet requirements has been allocated along with performance and other requirements to all necessary elements (e.g. hardware devices, software units, OS services, interfaces). Each IA requirement should be traced into all of the necessary design configuration items (or documented procedures, guidelines, etc.) responsible for demonstrating that requirement's functionality.

Once the requirements allocation has been validated, test and evaluation planning should be completed. IA testing inherently requires two types of validation techniques – that which can be demonstrated using standard techniques as part of functional system testing, and that which requires unique tests designed to stress potential IA vulnerabilities and risk areas.[13] Proper requirements traceability will take care of the bulk of IA requirements validation via functional testing, but if the IA design is to retain its stability, unique IA test events must take advantage of subsystem unit test windows and system integration and test periods. This will ensure that any required IA rework is

[13] "Improving Security Across the Software Development Life Cycle." National Cyber Security Partnership Task Force Reports. 1 Apr 04. 3 Jan 05. <http://www.cyberpartnership.org/SDLCFULL.pdf>

rolled into scheduled system find, fix and repair evolutions and facilitates the synchronization of IA configuration baselines with system configuration baselines.

4.3 Identifying and Codifying Recertification Expectations

In general, recertification requirements vary across C&A processes. What does not vary today is the amorphous nature of the events defined as requiring a recertification analysis. This is partly the result of the need to impart generic requirements appropriate to a wide range of systems, but is also partly due to the desire of most certification and accreditation authorities to assess system changes on a case-by-case basis. To adequately scope the resources and activities required to maintain a system over its entire life-cycle, however, more specific guidelines are necessary.

The program manager and security engineering personnel must work with the certification authorities to establish reasonable guidelines regarding what system functional changes, software updates or technology insertions will trigger recertification analyses, testing, C&A documentation updates, etc.

While the negotiated “triggers” may never be as specific as the security engineer desires, they should assist in the planning of system update cycles. This step is uniquely important in the tactical system design setting because change assessments are rarely required exclusively because of IA interests. Tactical systems often have weapons safety requirements, Human/Machine Interface standards, environmental qualification guidelines, interoperability rules, etc. that also require analysis and system recertification whenever changes are made. Documenting more specific IA recertification triggers will allow system designers to better coordinate and scope the resources required to recertify the system, and may even allow changes to be grouped in order to maximize the cost/benefit ratio of recertification activities.

5.0 System Development and Demonstration Phase

DAG Defined Acquisition / Systems Engineering Activities	Critical Information Assurance Design and Engineering Activities

• Key Inputs:
  ○ System Performance Specification
• Define Interface and Integration Requirements
• Develop Product Documentation
• Preliminary and Critical Design Reviews
• Fabricate Components / Code Software / Acquire Commercial Items
• Integration, Test & Evaluation
• Key Outputs:
  ○ Initial Product Baseline
  ○ Test Reports
	1. Ensure secure component and end-to-end system configurations
	
	2. Conduct initial residual risk quantification
	
	3. Ensure certification and accreditation authority participation in preliminary and final design reviews

5.1 Secure Component and End-to-End Configurations

As the system design effort moves through final design reviews and into formal evaluation phases, it is crucial that low-level IA-driven configuration activities are scheduled to complete prior to initial unit, component and integration tests. The IA requirements analysis will have identified the applicable Security Technical Implementation Guides (STIGs) the system is required to implement. There are DoD mandated STIGs for nearly every critical component of a system (e.g. OSs, network infrastructure, database and other applications, routers). As one of the primary means of enabling the core IA functionality within a system and plugging known vulnerabilities, the STIGs focus at the foundation levels of the system (e.g. OS services and settings, network support services, router settings) and can have a profound influence on how the rest of the system is engineered and integrated.

Compliance with all defined IA requirements will of course be necessary, but because these configuration requirements are so central to the overall IA performance of the system, and because compliance with the STIGs is often a key metric for certifying and accrediting authorities, it is necessary that there is a focused effort to verify these configurations. System scans or checklists utilized to validate STIG compliance must be done on a stable system configuration to ensure system functionality has been properly configured to perform within the constraints of the secure configuration. In other words, STIG implementation and compliance validation activities should not be conducted outside the mainstream design environment.

5.2 Initial Residual Risk Quantification

If security practitioners follow any of the standard DoD C&A processes faithfully, they will eventually conduct the tasks necessary to develop a Residual Risk report. What is too often the case, however, is that this assessment is scheduled too late in the system development process to support key decision points and security engineering tasks. The first of these analyses must be completed prior to finalizing the production configuration for several reasons:

- Tactical systems are frequently pressed into service prior to final validation testing due to operational demands. Without a risk assessment, the risk of operating the system in its as-is configuration will be unknown.
- More so than other types of systems, tactical systems often do not undergo the bulk of their integration and operability testing until after installation in their deployed environment. If the initial conduct of a residual risk assessment is not until after final installation, the gathering of interim or final IA approvals to operate can be significantly delayed.
- The residual risk analysis should be more than a formality, some risks may exceed the risk threshold of the certifying or accrediting authorities – if the system has left the development environment, rework can become time and resource intensive.

Ideally, an initial residual IA risk analysis should be available for review during a system's formal preliminary design review and updates presented prior to design approval at the critical design review.

5.3 Certification & Accreditation Authority Participation At System Design Reviews

Just as IA component configuration activities, testing and risk analyses must be conducted in conjunction with other systems engineering tasks, seats at the design review table must be reserved for certifying and accrediting authorities in addition to the program management, acquisition and user representatives normally in attendance. In general, incremental design reviews at all levels should include representation from IA certification and accreditation authorities.

There is often a strong desire to avoid exposing the system design at early stages or to avoid "airing dirty laundry" in front of management authorities of any kind. But because a robust IA implementation will permeate nearly every software module, network device, interface, technical manual, etc. built for the system, it is essential that the IA design is assessed incrementally throughout the design process.

Frequent socialization of the design with certification and accreditation representatives will also help them to realize they are actually partners invested in the design effort, not simply outsiders charged with periodically evaluating progress. Comprehensive reviews are especially important as the system nears its pre-production baseline, and test and evaluation evolutions are finalized.

6.0 Production and Deployment Phase

DAG Defined Acquisition / Systems Engineering Activities	Critical Information Assurance Design and Engineering Activities
• Key Inputs: ○ Test Results ○ Acquisition Program Baseline • Analyze Known Deficiencies and Identify Solutions • Interoperability Certification and Operability Testing • Key Outputs: ○ Production Baseline	1. Develop user-focused Information Assurance documentation and training 2. Establish regular Information Assurance readiness assessments

6.1 User-Focused Information Assurance Documentation and Training

Another important, but often neglected set of engineering activities is the development of documentation and training specifically focused on passing on IA operational details to the end-user. As has been noted previously, tactical systems and their users are often forward deployed to remote environments. This frequently means they are without dependable access to off-site technical assistance and must function self-sufficiently. In addition, as the technology integrated into today's tactical systems becomes ever-more complex, the IA functionality and the management tasks that go along with that functionality become more complex as well.

Despite those realities, system development efforts regularly shortchange the infusion of IA management and administration data into technical manuals, user's guides and IA-centric training associated with the installed system. Strong system access controls, intrusion detection systems, a secure router and real-time virus detection software for example are all of little utility if the personnel charged with security administration cannot review and backup audit logs, understand system IA alerts, respond to an IP address block request, or push anti-virus script updates to system workstations.

Virtually all DoD C&A processes require the development of user-focused IA documentation (e.g. Trusted Facility Manuals, Security Features User's Guides)[14], but they are usually developed with augmentation of the C&A package as the primary objective and with certifying and accrediting authorities as the intended audience. What system training curricula and support documentation must do, however, is give the end-users charged with security administration the knowledge and technical detail necessary to perform day-to-day IA management activities, respond to potential threat

[14] United States. Department of Defense Manual 8510.1-M, "DoD Information Technology Security Certification & Accreditation Process Application Manual." 31 Jul 00. 3 Jan 05. <http://www.dtic.mil/whs/directives/corres/html/85101m.htm>

activity or security incidents and maintain the certified configuration of the system. Validation of the suitability of these materials should be done with end-user participation and must incorporate their input.

6.2 Information Assurance Readiness Assessments

The maintenance of a strong system IA posture after delivery can only be guaranteed via regular stress testing. Software and other formal system updates, unauthorized local configuration changes, etc. all potentially alter IA performance. The systems engineering activities associated with updating the system baseline design must also include IA testing evolutions that re-validate overall IA system performance.

In addition, regular system grooms (prior to and again after operational deployments), penetration tests and vulnerability assessments will ensure the system is not only continually measured against changing threat and attack models, but will also support required periodic C&A recertification activities. These tasks must be scoped and notionally scheduled prior to the final baselining and production of the system to ensure they are tailored properly to the specific operational and environmental realities of the tactical system.

7.0 Operations and Support Phase

DAG Defined Acquisition / Systems Engineering Activities	Critical Information Assurance Design and Engineering Activities

• Key Inputs:
  ○ Service Use Data
  ○ User Feedback
• Execute Life-Cycle Support Program
• Evaluate System Modifications
• Test, Implement and Field Approved Modifications
• Key Outputs:
  ○ System Modifications
	1. Proactive system security administration and management

7.1 System Security Administration and Management

As mentioned previously, the foundation for system administration and management tasks must be established in advance of system deployment and operation, and then frequently revisited and revised. Change management will drive many IA-related management activities. IA relevant change forces come from many directions and in many different forms:

- Technology insertion to provide new functionality

- Technology refreshment to overcome component obsolescence
- Software updates to improve performance, patch vulnerabilities, repair faults, etc.

Regardless of the cause, the security engineer on the development side and the security administrator on the end-user side need to be conscious of the necessary steps required to re-validate the system's IA posture after changes are implemented. In essence, a survey of the steps covered in sections 3 through 6 above during change development should help identify the subset of tasks necessary to validate and re-certify (if necessary) the system's IA performance.

From an IA management standpoint, the necessary activities to adequately monitor and administer the system may be numerous and time intensive. The bottom line is that the security administrator must ensure he is given the tools necessary to do his job (both functionally within the system and via support documentation), obtains the requisite training to perform the required duties, and has the organizational authority needed to proactively manage the IA elements of the system.

8.0 Conclusion

It should be apparent, despite mature DoD IA requirements and C&A mandates, that without a substantive focus on systems security engineering activities the development of a tactical system with a robust IA posture is not guaranteed. This is especially true for tactical system development efforts where functional performance requirements are granted the vast majority of development resources and management attention.

With these realities as a given, it is essential that program personnel identify the necessary security engineering tasks required at each acquisition phase and ensure they are integrated seamlessly with mainstream system design and development activities. To achieve its objective, this guidebook has been developed to identify these tasks and make recommendations regarding how to maximize their benefit. The warfighters that depend on the tactical systems we provide them deserve systems infused with the best IA functionality that can be delivered.

References

Defense Acquisition Guidebook Web Page. 2004. United States Department of Defense. 3 January 2005. <http://akss.dau.mil/dag/DoD5000.asp?view=document>

"DoD Publishes Defense Acquisition Guidebook," DefenseLINK News Release No. 1025-04. 14 October 2004. 3 January 2005. <http://www.defenselink.mil/releases/2004/nr20041014-1389.html>

Harris, Shon. All-In-One CISSP Certification Exam Guide. Berkely: Mcgraw-Hill/Osborne, 2002.

"Improving Security Across the Software Development Life Cycle." National Cyber Security Partnership Task Force Reports. 1 April 2004. 3 January 2005. <http://www.cyberpartnership.org/SDLCFULL.pdf>

Moore, Andrew P. "Analyzing the Threat Dynamics of Complex Networked Systems." Carnegie Mellon 2004 CyLab Research Projects. 3 January 2005. <http://www.cylab.cmu.edu/default.aspx?id=282>

Pfleeger, Charles P. Security in Computing, 2nd Edition. Upper Saddle River: Prentice-Hall, 1997.

United States. Department of Defense Directive 8500.1, "Information Assurance." 24 October 2002. 3 January 2005. <http://www.dtic.mil/whs/directives/corres/pdf/d85001_102402/d85001p.pdf>

United States. Department of Defense Instruction 5000.2, "Operation of the Defense Acquisition System." 12 May 2003. 3 January 2005. <http://akss.dau.mil/dag/DoD5000.asp?view=document&doc=2>

United States. Department of Defense Instruction 8500.2, "Information Assurance Implementation." 6 February 2003. 3 January 2005. <http://www.dtic.mil/whs/directives/corres/pdf/i85002_020603/i85002p.pdf>

United States. Department of Defense Instruction 8580.1, "Information Assurance in the Defense Acquisition System." 9 July 2004. 3 January 2005. <http://www.dtic.mil/whs/directives/corres/html/85801.htm>

United States. Department of Defense Manual 8510.1-M, "DoD Information Technology Security Certification & Accreditation Process Application Manual." 31 July 2000. 3 January 2005. <http://www.dtic.mil/whs/directives/corres/html/85101m.htm>

United States. National Security Telecommunications and Information Systems Security
Policy No. 11, Revised Fact Sheet, "National Information Assurance Acquisition Policy." July 2003. 3 January 2005. <http://niap.nist.gov/cc-scheme/nstissp_11_revised_factsheet.pdf>

Wood, Bradley J. "An Insider Threat Model for Adversary Simulation." SRI International, Cyber Defense Research Center, System Design Laboratory Publications. 12 July 2000. 3 January 2005. <http://www.csl.sri.com/users/bjwood/Insider_threat_model_v02.pdf>

Last Updated: June 15th, 2019

Upcoming SANS Training

Click here to view a list of all SANS Courses

Security Operations Summit & Training 2019	**New Orleans, LAUS**	**Jun 24, 2019 - Jul 01, 2019**	**Live Event**
SANS ICS Europe 2019	**Munich, DE**	**Jun 24, 2019 - Jun 29, 2019**	**Live Event**
SANS Cyber Defence Japan 2019	**Tokyo, JP**	**Jul 01, 2019 - Jul 13, 2019**	**Live Event**
SANS Paris July 2019	**Paris, FR**	**Jul 01, 2019 - Jul 06, 2019**	**Live Event**
SANS Munich July 2019	**Munich, DE**	**Jul 01, 2019 - Jul 06, 2019**	**Live Event**
SEC450 Security Ops-Analysis Beta 1	**Crystal City, VAUS**	**Jul 08, 2019 - Jul 13, 2019**	**Live Event**
SANS London July 2019	**London, GB**	**Jul 08, 2019 - Jul 13, 2019**	**Live Event**
SANS Pittsburgh 2019	**Pittsburgh, PAUS**	**Jul 08, 2019 - Jul 13, 2019**	**Live Event**
SANS Cyber Defence Singapore 2019	**Singapore, SG**	**Jul 08, 2019 - Jul 20, 2019**	**Live Event**
SANS Charlotte 2019	**Charlotte, NCUS**	**Jul 08, 2019 - Jul 13, 2019**	**Live Event**
SANS Rocky Mountain 2019	**Denver, COUS**	**Jul 15, 2019 - Jul 20, 2019**	**Live Event**
SANS Columbia 2019	**Columbia, MDUS**	**Jul 15, 2019 - Jul 20, 2019**	**Live Event**
SANS Pen Test Hackfest Europe 2019	**Berlin, DE**	**Jul 22, 2019 - Jul 28, 2019**	**Live Event**
SANS San Francisco Summer 2019	**San Francisco, CAUS**	**Jul 22, 2019 - Jul 27, 2019**	**Live Event**
DFIR Summit & Training 2019	**Austin, TXUS**	**Jul 25, 2019 - Aug 01, 2019**	**Live Event**
SANS Riyadh July 2019	**Riyadh, SA**	**Jul 28, 2019 - Aug 01, 2019**	**Live Event**
SANS Boston Summer 2019	**Boston, MAUS**	**Jul 29, 2019 - Aug 03, 2019**	**Live Event**
SANS July Malaysia 2019	**Kuala Lumpur, MY**	**Jul 29, 2019 - Aug 03, 2019**	**Live Event**
SANS Crystal City 2019	**Arlington, VAUS**	**Aug 05, 2019 - Aug 10, 2019**	**Live Event**
SANS Melbourne 2019	**Melbourne, AU**	**Aug 05, 2019 - Aug 10, 2019**	**Live Event**
Security Awareness Summit & Training 2019	**San Diego, CAUS**	**Aug 05, 2019 - Aug 14, 2019**	**Live Event**
SANS London August 2019	**London, GB**	**Aug 05, 2019 - Aug 10, 2019**	**Live Event**
SANS Prague August 2019	**Prague, CZ**	**Aug 12, 2019 - Aug 17, 2019**	**Live Event**
Supply Chain Cybersecurity Summit & Training 2019	**Arlington, VAUS**	**Aug 12, 2019 - Aug 19, 2019**	**Live Event**
SANS Minneapolis 2019	**Minneapolis, MNUS**	**Aug 12, 2019 - Aug 17, 2019**	**Live Event**
SANS San Jose 2019	**San Jose, CAUS**	**Aug 12, 2019 - Aug 17, 2019**	**Live Event**
SANS MGT516 Beta Three 2019	**Arlington, VAUS**	**Aug 19, 2019 - Aug 23, 2019**	**Live Event**
SANS Amsterdam August 2019	**Amsterdam, NL**	**Aug 19, 2019 - Aug 24, 2019**	**Live Event**
SANS Virginia Beach 2019	**Virginia Beach, VAUS**	**Aug 19, 2019 - Aug 30, 2019**	**Live Event**
SANS Chicago 2019	**Chicago, ILUS**	**Aug 19, 2019 - Aug 24, 2019**	**Live Event**
SANS August Malaysia 2019	**Kuala Lumpur, MY**	**Aug 19, 2019 - Aug 24, 2019**	**Live Event**
SANS Tampa-Clearwater 2019	**Clearwater, FLUS**	**Aug 25, 2019 - Aug 30, 2019**	**Live Event**
SANS Cyber Defence Canberra 2019	**OnlineAU**	**Jun 24, 2019 - Jul 13, 2019**	**Live Event**
SANS OnDemand	**Books & MP3s OnlyUS**	**Anytime**	**Self Paced**