The United States Department of Defense Revitalization of System Security Engineering Through Program Protection

Kristen Baldwin
Principal Deputy, DASD, Systems Engineering
United States Department of Defense
Washington, DC USA

John F. Miller
The MITRE Corporation
McLean, VA USA

Paul R. Popick
The Aerospace Corporation
Chantilly, VA USA

Jonathan Goodnight
Dynamics Research Corporation
Arlington, VA, USA

Abstract— **This paper discusses the U.S. Department of Defense (DoD) state-of-practice of integrating security into systems engineering (SE) in order to implement system security engineering (SSE) through the program protection process. The discussion includes a description of new policies and the application of methods and techniques to implement SSE. Although SSE is normally viewed as a specialty engineering area, this paper emphasizes the need to more tightly integrate SSE with the overall systems engineering.**

Keywords-system security; security engineering; system security engineering; program protection; system assurance; software assurance; secure design; secure software; supply chain risk management; supply chain integrity

I. INTRODUCTION

This paper discusses the U.S. Department of Defense (DoD) state-of-practice for integrating security into systems engineering (SE) in order to implement system security engineering (SSE) through the program protection process. The discussion includes a description of new policies and the application of methods and techniques to implement SSE. Although SSE is normally viewed as a specialty engineering area, this paper emphasizes the need to more tightly integrate SSE with overall systems engineering.

According to the DoD Military Handbook *System Security Engineering Program Management Requirements,* SSE is "an element of system engineering that applies scientific and engineering principles to identify security vulnerabilities and minimize or contain risks associated with these vulnerabilities. It uses mathematical, physical, and related scientific disciplines, and the principles and methods of engineering design and analysis to specify, predict, and evaluate the vulnerability of the system to security threats" [1].

II. BACKGROUND

Three trends are contributing to the system security challenges facing major DoD programs. First, DoD systems are increasingly relying on commercially available technology that is frequently developed and manufactured outside of U.S. control and, perhaps more important, is widely available for all the world to study, reverse engineer, and identify vulnerabilities [2]. Second, the complex supply chains of major acquisition programs (prime contractors, subcontractors, suppliers, and sub-suppliers) make it difficult for anyone to truly know what is in the system and where it came from. In 2006, the Director of Logistics for the Joint Staff reported that the Department had more than 100,000 active suppliers serving more than 30,000 DoD customers worldwide [3]. Third, system complexity and interconnectedness (via software dependence and connections to numerous DoD networks) obfuscate the possible system states and vulnerabilities. This challenge is tied to the growing software-intensive nature of systems. A 2006 Software Industrial Base Assessment found that the amount of software used in individual DoD weapon systems has grown exponentially. As an example, the study highlighted the growth in fighter aircraft functionality requiring software from 8% in the F-4 in 1960 to 80% in the F-22 in 2000 [4]. Taken together, these trends are decreasing the Department's confidence that its systems will function as intended.

III. NEW ADVANCES IN SSE POLICY, ACTIONABLE GUIDANCE, AND BEST PRACTICES

A. Program Protection Planning

The Department is extending its program protection process to apply SSE principles to defense acquisition programs. The process and its artifact, the Program Protection Plan (PPP), historically emphasized protecting advanced research and technology from unauthorized or inadvertent disclosure; if a weapon system derived a capability advantage from some leading-edge technology, algorithm, or component, that critical program information (CPI) was closely guarded through process and technical means. However, this approach was not securing the supply chain from alteration or substitution risk to mission-critical system elements that used commercially

available technology, and it did not consider unintentional system design vulnerabilities. Today's drone systems are built using a combination of COTS and DoD-unique hardware and software. In the past, the DoD was primarily focused on protecting the advanced technology (as might be found in the drone's sensors) from disclosure, but the drone must also be protected from insertion of backdoors, worms, and malicious acts in the supply chain; and, the system design must be defended against unauthorized access, control, or alteration during operations.

In addition, perimeter security techniques engendered by information systems security engineering and other disciplines lack sufficient point defense for system critical component functions should a perimeter defense be compromised; this is increasingly problematic as systems grow in complexity and adversaries have more opportunities for entry. Finally, the PPP was not required until Milestone B, at which point most of the design was fixed and the addition of new security features frequently brought negative cost and schedule impacts.

Recent DoD issuances and regulatory changes addressing some of these structural issues and providing the framework for implementing more rigorous SSE include:

- In 2008, the Under Secretary of Defense for Intelligence issued new CPI protection policy requiring Milestone Decision Authority approval of the PPP [5].
- In 2009, the Under Secretary of Defense for Acquisition, Technology, and Logistics (USD(AT&L)) and the DoD Chief Information Officer approved the DoD Trusted Defense Systems Strategy in response to FY2009 National Defense Authorization Act (NDAA) Section 254 [6].
- In 2010, the Deputy Secretary of Defense issued a Directive-Type Memorandum establishing policy and pilot activities for supply chain risk management with an emphasis on information and communications technology (ICT) components [7].
- In 2011, the Principal Deputy, USD(AT&L) issued a memorandum requiring every acquisition program to complete a PPP at Milestone A and to update it at each subsequent milestone and the Full-Rate Production (FRP) decision. The policy also required every program, even those without CPI, to identify mission-critical functions and components and manage their risk of compromise [8].

In 2012, USD(AT&L) will publish an update to the Defense Acquisition Guidebook with a new chapter on program protection planning that includes guidance on SSE entry and exit criteria for Systems Engineering Technical Reviews (SETRs) throughout the acquisition life cycle. Fig. 1 illustrates the DoD opportunities to emphasize SSE via the PPP and SETR engagement points throughout the Defense Acquisition System life cycle.

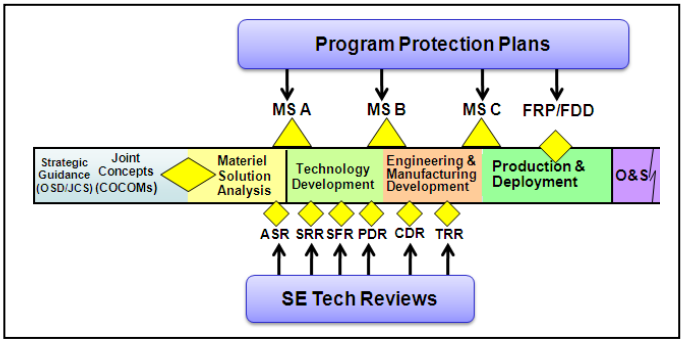

Figure 1. SSE engagement opportunities throughout the acquisition life cycle

B. *Prioritization: Identifying Mission-Critical Functions and Components Through Criticality Analysis*

Risk management is a core tenet of SSE, and it drives the prioritization of which systems and which elements within those systems will be protected by countermeasures. System prioritization criteria include the Mission Assurance Category (an indication of the confidentiality and availability requirements), National Security System status, and other factors reflecting the relative warfighting importance of the platform. This prioritization is reflected in the acceptable level of risk to the functionality of that system.

Within a given system, mission-critical functions are those functions of the system that, if corrupted or disabled, would unacceptably degrade the system effectiveness in achieving the core mission for which it was designed. Mission-critical components are the system elements (hardware, software, and firmware) that implement critical functions, including components that defend or have unmediated access to intrinsically critical components.

Criticality Analysis (CA) is the DoD method by which mission-critical components are identified and prioritized. It is an end-to-end functional decomposition of the system that involves identifying mission threads, decomposing them into their mission-critical functions and components, and prioritizing them by assigning Criticality Levels as defined in Table 1. CA and the protection failure levels are similar in theory to the Failure Modes, Effects, and Criticality Analysis (FMECA) used for reliability or safety, albeit with a smaller level of effort and consideration of malice in addition to natural hazards. The general CA steps are:

- Group mission capabilities by relative importance (e.g. "Fire Control" may be more important than "Report")
- Decompose mission capabilities into critical functions (e.g. Fire Control mission may consist of Target, Acquire, Fix, Track, Fire, and Confirm functions)
- Map missions and critical functions to critical components using system architecture diagrams, fault tree analyses, requirements matrices, or other technical documents.
- Identify and include any components that do not directly implement critical functions, but either have unmediated access to or protect critical functions
- Assign Criticality Levels to the identified critical components; Heuristics can include redundancy (which can indicate concern that a function is performed) and frequency of use across mission threads.

TABLE I. PROTECTION FAILURE CRITICALITY LEVELS

Level I – Total Mission Failure
Level II – Significant/Unacceptable Degradation
Level III – Partial/Acceptable Degradation
Level IV – Negligible

The CA is led by systems engineers and mission/operator representatives; however, it is a team effort and mission-critical functions and components must be identified by a multi-disciplined group.

Early in the life cycle, it may be possible to execute the steps only at a "high level" or to complete only some of the initial steps in the CA process. For example, the first pass through the CA may identify the critical functions but none of the potential implementing components. The CA is thus an iterative process and, to be effective, it must be executed throughout the acquisition life cycle. Through the integration of systems engineering into the program protection process, DoD is requesting updated CAs at each SETR, building on the growing system maturity, incorporation of security designs, knowledge gained from prior CAs, updated risk assessment information, and updated threat and vulnerability data.

C. *Threats, Vulnerabilities, and Countermeasures*

The CA provides a foundation for assessing and prioritizing threats to the system, vulnerabilities to those threats, and countermeasures for mitigating those vulnerabilities. Fig. 2 shows how the results of the CA, vulnerabilities assessment (VA), threat assessment (TA) and countermeasure alternatives are brought together as the basis for a risk-cost-benefit trade-off. In the risk cube, the Consequence factor is determined from the CA levels and the Likelihood factor is determined from the VA and TA. Possible countermeasures are evaluated relative to the risk level, the acceptable risk, and the implementation cost. Because vulnerabilities are constantly discovered, an iterative reassessment and trade-off is needed through the acquisition and sustainment of the system. Details on threats, vulnerabilities, and countermeasures are highlighted in the remainder of this section.

1) *Threats*

DoD systems are exposed to threats of malicious insertion and tampering throughout the development and supply of critical components. This exposure is further exacerbated by the use of a significant number of commercial-off-the shelf (COTS) supplied parts that are obtained through a global supply chain. Examples of malicious insertion threats are widely publicized and include such events as radar systems that are unable to detect a particular country's planes and the disabling of centrifuges through the "stuxnet" computer worm.

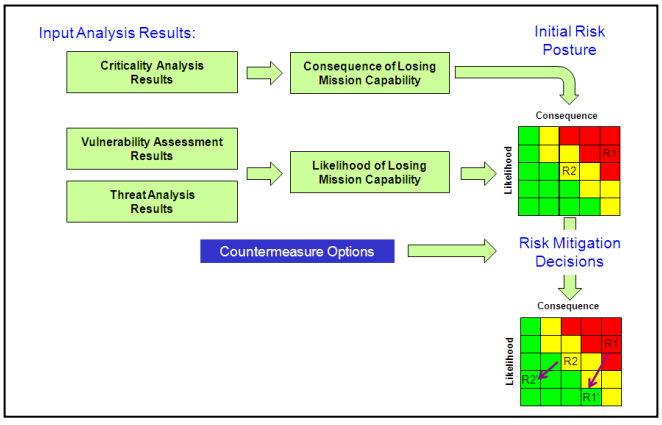

Figure 2. Risk-cost-benefit trade-off in the program protection process

2) *Vulnerabilities*

A vulnerability is any weakness in system design, development, production, or operation that can be exploited by a threat to defeat a system's mission objectives or significantly degrade its performance. Vulnerabilities to watch for in the systems engineering processes include:

- Access paths within the supply chain that allow threat actors to introduce components that cause the system to fail at some later time ("components" here include hardware, software, and firmware); and
- Access paths that allow threat actors to trigger a component malfunction or failure at a time of their choosing.

"Supply chain" includes any point in a system's design, engineering and manufacturing development, production, test and evaluation, configuration in the field, updates, and maintenance.

Vulnerabilities are assessed with respect to exposure and exploitability. For example, a vulnerability may expose a critical function in a way that has only a "marginal" consequence, while another vulnerability may expose a critical function in a way that has "catastrophic" consequence. Exploitability assesses the ease or difficulty of taking advantage of a vulnerability, the developers' or maintainers' ability to detect access used to introduce or trigger a vulnerability, and any other deterrents to threats such as the consequences of being caught.

3) *Countermeasures*

Countermeasures are cost-effective activities and attributes to mitigate or neutralize threats to and vulnerabilities of system functions and components. They vary from process activities to design attributes, and their implementation cost and system performance trade-offs must be considered against the criticality of the functions they would be protecting. The following subsections describe three particularly relevant classes of SSE countermeasures.

a) *Assurance-Focused Design*

Implementing countermeasures through assurance-focused designs along with defense in depth are fundamental elements of SSE. Examples of assurance-focused designs include use of separation kernels and diverse redundancy for critical functions and the establishment of secure design and code standards that address the riskiest vulnerabilities. A separation kernel [1] approach is an excellent way to protect Level I critical functions (those with total mission failure implications) identified by the mission CA. Separation kernels isolate critical functions and ensure a well-defined security policy definition of the relationships between the isolated critical functions and all other functions. The defense-in-depth approach encourages the use of multiple assurance mechanisms to reduce the impact when a vulnerability is exploited. For example, the use of least privilege for interaction between critical functions along with a

[1] A separation kernel creates an environment in which information can only flow along explicitly defined communication channels.

separation kernel for the interaction between critical functions and other functions is an example of a defense-in-depth technique that may reduce the impact of a supply chain insertion of malware. The use of fault isolation for policy violations to trap and recover from these violations, as well as to initiate alerts or authority changes to prevent additional violations, further extends the defense-in-depth approach.

Diverse redundancy builds upon the redundancy approach used for reliability by further requiring the use of functionally equivalent, physically diverse components obtained from different sources. This added complexity greatly increases the level of effort required to compromise a critical function by distributing trust across multiple supply chains.

b) Software Assurance

Software assurance is "level of confidence that software is free from vulnerabilities, either intentionally designed into the software or accidentally inserted at anytime during its life cycle and that the software functions in the intended manner" [9]. The FY2011 NDAA required DoD to develop a strategy for ensuring that software assurance is considered throughout the acquisition life cycle [10]. Through program protection, DoD is emphasizing the security of software development, procurement, testing, operation, and maintenance.

The provenance of software and the access to software need to be understood as the software moves through the procurement or development life cycle. Assurance-focused design requires awareness of common vulnerabilities, assessment of the specific vulnerabilities of COTS and legacy software, and the implementation of design features to address these vulnerabilities. A starting point is the use of wrappers for the COTS and legacy software. The wrappers would address common vulnerabilities identified by using databases such as the Common Vulnerabilities and Exposures (CVE) dictionary and specific vulnerabilities identified by static analyzers and inspection of the systems' software [11]. Including inspection for secure design and coding standards in the design and code reviews ensures that known vulnerabilities are avoided or reduced. Even the development environment tools used to create and translate the software into the executing system need to be assessed and protected. For example, the origin of library routines loaded into the system often have unknown or open source provenance that exposes the system to unknown vulnerabilities.

c) Supply Chain Risk Management

Countermeasures for protection of the supply chain require the use of a set of practices early in the system design to evaluate potential parts for supply chain vulnerabilities while design changes and component substitutions are still relatively easy. It becomes an essential systems engineering practice to include the evaluation of supply chain threats and vulnerabilities as part of the design trade space. While it is normal to trade off performance versus reliability and even some aspects of security, supply chain trade-offs are not typically included.

Supply chain risk management (SCRM) needs to be incorporated into the system development life cycle whenever designs and implementations are evaluated and alternative parts are considered. Currently the full end-to-end supply chain is not well understood. Many companies will know their first-tier suppliers but will not know the "subtier" suppliers (the suppliers that supply their suppliers). To fully understand the supply chain risks and vulnerabilities, the full supply chain must be understood for at least the most critical function components (i.e., Level I critical functions). Some of the barriers to a full understanding of the critical function/component supply chain include internal procurement systems that are unable to maintain the identification of the full supply chain (supplier and subtier suppliers) and a company's need to protect intellectual property by not revealing the supply chains for fear of being circumvented. [10] provides some legal authority for gathering and assessing information relating to supply chain risk to address some of these issues.

As the supply chain for the critical function components is being selected, the vulnerabilities respective to each supplier and their suppliers need to be understood and trade-offs made between alternative supply chain risks. Companies need to track the "provenance" of the critical function components as they move through the supply chain and understand the engagement points for the component. Supply chain countermeasures include the use of secure transportation, limiting access, tracking of all access, penetration testing and blind buys (i.e., where the acquirer identity is not revealed). In some cases the establishment or use of a trusted foundry[2] for the most critical of the Level I parts may be required. The DoD SCRM Key Practices and Implementation Guide contains a comprehensive treatment of possible supply chain countermeasures [12].

D. SSE Implementation Throughout the Acquisition Life Cycle

Fig. 1 shows the DoD acquisition life cycle phases with the milestones (MS A, MS B, MS C) that end each phase and the SETRs within each phase. The SETRs provide gates within the phase to assess progress and are used as indicators of readiness to move to the next set of activities. The system performance specification and design activities are well defined for each of the phases and technical reviews within the phase. [13] [14].

The key SSE program protection process activities and criteria described in the following sections are now being defined as an iterative SSE pattern. As discussed earlier, the primary activities of the program protection process are a CA, a TA, a VA, and the selection of countermeasures (mitigations) based upon a risk-cost–benefit trade-off, as shown in Fig. 2.

Each iteration of the SSE program protection process incorporates the SSE analysis results, including design countermeasures, from the previous iteration until an acceptable balance of risk, cost and benefit (protection) is achieved. At the same time, new or different threats and vulnerabilities may emerge as a result of the evolving system design or from updates coming from the external environment. While the primary SSE activities and criteria repeat for each of the SETRs and phase milestones, they evolve and their

[2] The Defense Microelectronics Activity accredits integrated circuit design, aggregation, mask and wafer fabrication, packaging, and test service providers as trusted, based on minimum personnel and process criteria.

emphasis and granularity change as the system design matures. In the following sections the emphasis of the key SSE activities and criteria are summarized for each phase and are discussed for each SETR in the context of the applicable phase.

1) Pre-Milestone A

During the Materiel Solution Analysis (MSA) phase that culminates with the Milestone A decision, the "system need" is established and high-level system requirements are defined. A notional system architecture is often created to assist with the requirements analysis and definition for the preferred system concept. The Alternative Systems Review (ASR) is conducted to ensure that the resulting set of requirements agrees with the customer needs and expectations. This is also an important review for evaluating system design and affordability trades.

During the MSA phase, mission threads are evaluated for critical mission capabilities and the CA identifies associated mission critical functions and establishes their criticality levels. Generic threat and vulnerability information is available. Specific vulnerabilities and supply chain threats may be obtained by examining the notional architecture that identifies potential components for selected Level I and II critical functions. As part of design concept trade-offs associated with potential alternative(s), countermeasures are analyzed to minimize vulnerabilities, weaknesses, and implementation costs, and these results inform the ASR. Risks can be assessed based upon past systems of a similar type, concept prototypes that may proceed into the technology development phase, the generic and identified specific threats and vulnerabilities, and supply chain considerations such as assessing potential suppliers for identified potential components. A set of countermeasure alternatives are developed, evaluated, and traded off with respect to the risks identified, the estimated cost, and the potential risk reduction achieved.

Together with other interim SE requirements analysis results, the selected SSE countermeasures are synthesized into the developing system requirements and the notional architecture. The SSE program protection process activities are iterated along with the various other SE requirements analysis activities until a suitable balance of system performance, risk, cost and security is achieved.

The SSE analysis and resulting requirements are discussed as part of the ASR and used for the Request for Proposal (RFP) for the next phase. Some of the supply chain countermeasures are incorporated into the RFP Statement of Work (SOW) because they are process requirements; others are incorporated into the system requirements. One best practice is to include in the RFP is the requirement to deliver Bill of Materials to the Department in order to make visible the complete supply chains of critical components. As early in the lifecycle as possible, RFPs will highlight need to procure parts from trusted suppliers; specifically DoD unique application specific integrated circuits (ASICs). The DoD and Intelligence community established a trusted foundry program in 2003 that provides trusted supply of leading edge ASICs to DoD.

2) Pre-Milestone B

During the Technology Development (TD) phase that culminates with the Milestone B decision, the system draft requirements are established through successful completion of the Systems Requirements Review (SRR). The requirements are baselined (approved) and the draft functional design is established upon successful completion of the System Functional Review (SFR). The functional design is baselined and the draft allocated design is established through successful completion of the Preliminary Design Review (PDR).

During the TD phase, the common set of SSE activities (CA, VA, TA, risk assessment and countermeasures selection) are iterated before each SETR, taking into account updated threat data to determine whether additional countermeasures are necessary. The threat data is refreshed based upon the evolving functional and allocated designs. The list of mission critical functions, criticality levels, and implementing critical components is refined and updated, with a focus on logic bearing elements. The known and potential components that implement the Level I and II critical functions are examined for additional exploitable vulnerabilities. Vulnerabilities are identified through analysis of system component interactions and the use of databases, such as CVE, which catalog known software vulnerabilities. Residual risks (remaining after countermeasure implementation) are assessed and addressed.

Prototyping efforts and design trade-offs for risk reduction during the TD phase include minimizing the attack surface and employing affordable, risk-based countermeasures. The emerging (preliminary) design countermeasures will include software assurance considerations such as the use of fail-safe defaults, defense in depth, purging of temporary data, use of secure languages and libraries, avoidance of unsafe coding constructs, and use of static code analysis.

The system requirements, functional design, and allocated design are updated as necessary to incorporate any new countermeasures selected. Countermeasure effectiveness is evaluated at each level of the maturing design. The SSE subset of the requirements traceability verification matrix (RTVM) is examined to ensure that all SSE requirements have been allocated to the design. The logical design that implements the SSE requirements and the allocated design countermeasures are incorporated into the RFP for the next phase, and SSE process countermeasures are included in the SOW. Supply chain considerations such as the use of diverse redundancy and blind buying strategies are important during this analysis to refine the system detailed design.

3) Pre-Milestone C

During the Engineering and Manufacturing Development (EMD) phase that culminates with the Milestone C decision, detailed system design is completed and the system capability and manufacturability process are demonstrated. Through a successful completion of the Critical Design Review (CDR), the design release is approved and the initial product baseline is established. A successful Test Readiness Review (TRR) establishes readiness of the system to begin acceptance testing, and a successful Production Readiness Review (PRR) establishes readiness for manufacturing and limited production.

The common set of SSE activities is again iterated during the EMD phase in order to finalize the list of critical components and subcomponents in the detailed design. Residual risks are reassessed, high-priority mitigations are tracked, and the plans for associated countermeasures (both

process and design related) are updated. Before specific subcomponent selection (hardware parts and software modules) is finalized, updated component threats and associated countermeasures are evaluated with respect to cost-benefit-risk to balance security considerations with the system performance and cost. Design countermeasures for implementation, testing, and operational use are refined and include software assurance considerations. Identification of associated process and design countermeasures is supported by databases such as CVE to identify vulnerabilities that enable various types of attacks; Common Attack Pattern Enumeration and Classification (CAPEC) to analyze environments, code, and interfaces for common destructive attack patterns; and Common Weakness Enumeration (CWE) to examine software architectures, designs, and source code for weaknesses [15][16]. Software analysis tools are employed to assist in common vulnerability detection.

Subcomponents are assessed to ensure that throughout production, deployment, operations, and sustainment, they will provide effective protection for associated critical functions. A comprehensive, end-to-end test approach is developed to verify system-level security requirements as documented in the RTVM, including verification of specific countermeasures allocated to subcomponents in the detailed baselines.

4) Full-Rate Production and Beyond

During the Production and Deployment phase, the common set of SSE activities is iterated to review and maintain the list of critical system components and subcomponents, to ensure appropriate coverage in configuration audits and to ensure life cycle integrity throughout operations and sustainment. Activities focus on supply chain integrity and ensuring that any further software coding, hardware fabrication, and system integration tasks provide the required level of system security. Security threat and attack scenarios are updated, as are the associated supply chain process countermeasures included in production planning and the system operational concept.

Life cycle sustainment planning includes security requirements for system upgrades and COTS component procurements throughout the anticipated life of the system. The plans define the period and the events which trigger another iteration of the SSE activities. Because the threat may change significantly as suppliers merge or change sourcing strategies over a very short period of time, SSE activities need to be repeated every 18 to 24 months during sustainment, as well as whenever a significant change of components occurs.

IV. Future Work

DoD has a University Affiliated Research Center, called the Systems Engineering Research Center (SERC), which is a consortium of universities led by the Stevens Institute of Technology and the University of Southern California Information Sciences Institute to perform studies for the DoD. The SERC produced an SSE research roadmap in 2010 and is now performing multiple research tasks from that roadmap, including an effort to prototype SSE design techniques and estimate the system trade-offs (cost, latency, power, etc.) that must be made to implement them [17][18]. In addition, DoD is working with the International Council on Systems Engineering (INCOSE) to update the INCOSE Systems Engineering Handbook with SSE considerations.

In addition to refining the methods, processes, and tools for SSE, the DoD is identifying opportunities for better integration with the test and evaluation and sustainment communities. If SSE process controls and design features are not exercised in developmental and operational test plans and procedures, the planning and engineering efforts could be wasted if vulnerabilities go undiscovered. Similarly, maintaining and repairing secured systems with unsecured processes and components will introduce sources of vulnerability outside the acquisition process.

More research and development is needed to produce engineering methods and tools to execute the system security engineering process as outlined in this paper. Ideally, this engineering specialty will leverage existing engineering methodology; building a security lens into reliability and safety practices, for example. Industry becomes a critical partner in achieving SSE as well, and as the requirement for security appears more frequently in RFPs, it is expected that the defense industrial base will build secure design practices into their engineering functions.

References

[1] *System Security Engineering Program Management Requirements*, MIL-HDBK-1785, 1995.

[2] M. F. McGrath, *et al*., "Equipping Tomorrow's Military Force: Integration of Commercial and Military Manufacturing in 2010 and Beyond," Board on Manufacturing and Engineering Design, National Research Council, National Academies Press, Washington, DC, 2002.

[3] LTG C. V. Christianson, "DoD Supply Chain as a System of Systems," presented at the 2nd Annual System of Systems Engineering Conference, Fort Belvior, VA, 2006.

[4] P. Chao, "An Assessment of the National Security Software Industrial Base," Center for Strategic and International Studies, Washington, DC, 2006.

[5] *Critical Program Information (CPI) Protection Within the Department of Defense*, DoD Instruction 5200.39, 2008.

[6] *Report on Trusted Defense Systems in Response to National Defense Authorization Act, Section 254*, DoD Congressional Report, 2009.

[7] *Supply Chain Risk Management (SCRM) to Improve the Integrity of Components Used in DoD Systems*, Directive-Type Memorandum 09-016, 2010.

[8] F. Kendall, "Document Streamlining–Program Protection Plan (PPP)," Principal Deputy Under Secretary of Defense for Acquisition, Technology, and Logistics, Washington, DC, 2011.

[9] *National Information Assurance (IA) Glossary*, Committee on National Security Systems Instruction No. 4009, 2010.

[10] Ike Skelton National Defense Authorization Act for Fiscal Year 2011, Public Law No. 111-383, sections 806 and 932, 2011.

[11] The Mitre Corporation, "Common Vulnerabilities and Exposures; The Standard for Information Security Vulnerability Names." Internet: http://cve.mitre.org [Jan. 23, 2012].

[12] *Key Practices and Implementation Guide for the DoD Comprehensive National Cyber Initiative 11 Supply Chain Risk Management Pilot Program, February 25, 2010,* SCRM Program Management Office, Global Task Force, OSD(NII)-CIO/ODASD(CIIA).

[13] *Operation of the Defense Acquisition System*, DoD Instruction 5000.02, 2008.

[14] Defense Acquisition University, Defense Acquistion Guidebook, Chapter 4, "Systems Engineering." Internet: https://acc.dau.mil/communitybrowser.aspx?id=332951 [Jan. 23, 2012]

[15] R. Jones and B. Horowitz, “System-Aware Cyber Security”, ITNG, 2011 Eighth IEEE International Conference on Information Technology: New Generations, April 2011, p. 914-917.

[16] The Mitre Corporation, “Common Weakness Enumeration; A Community-Developed Dictionary of Software Weakness Types.” Internet: http://cwe.mitre.org [Jan. 23, 2012].

[17] The Mitre Corporation, “Common Attack Pattern Enumeration and Classification; A Community Knowledge Resource for Building Secure Software.” Internet: http://capec.mitre.org [Jan. 23, 2012].

[18] J. Bayuk, *et al.*, “System Security Engineering: Final Technical Report SERC-2010-TR-005,” Systems Engineering Research Center, Stevens Institute of Technology, Hoboken, NJ, 07030.

[19] National Defense Industrial Association, “Engineering for System Assurance,” Version 1.0, 2008.