The Buck Stops Here: Systems Engineering is Responsible for System Security

Rick Dove, rick.dove@incose.org
Paul Popick, paul.popick@incose.org
Beth Wilson, elizabeth.wilson@incose.org

Who is responsible for systems security? As shown in figure 1, the acquirer (Acq) thinks it is the supplier, the supplier (Sup) delegates that responsibility to systems engineering, who pass it on to system security engineering (SSE), who meet requirements originating with the acquirer. This arrangement results in a finger-pointing circle when security fails.

New revisions to the INCOSE *Systems Engineering Handbook* are integrating responsibility for system security into the systems engineering processes. Placing responsibility on systems engineering is only a first step. A second step requires mutual engagement between systems engineering and security engineering, an engagement that can only be enabled by systems engineering. Systems engineers and program or project managers will be expected to engage effectively throughout the systems engineering processes and activities—beginning with requirements analysis and the concept of operations, and proceeding through the full lifecycle of development, operations, and disposal.

The theme articles in this issue of *INSIGHT* focus on the nature and problems of effective security engineering engagement in critical systems engineering processes. In the end, the acquirer and the supplier must also engage, in a shared responsibility that recognizes and deals with an unpredictable future of security threats. But that is another story, one that cannot be effective until systems and security engineering engagement is achieved.

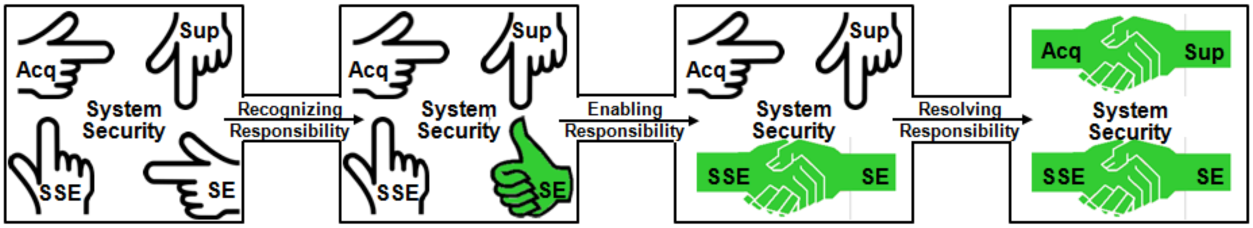

Figure 1. Effective evolution of systems security responsibility begins with systems engineering

As systems engineers we find that our systems are under attack by intelligent and innovative adversaries. We rely on the specialty engineering function of system security engineering to protect what we design, build, and deploy. Unfortunately the results are not encouraging. The costs invested in system security engineering and deployed security technology are increasing every year, while the losses caused by breaches are also increasing every year. Something is not right. Something needs to be done differently.

In government acquisition projects, systems engineering security concerns are driven by security requirements specified by the customer, generally in the form of adherence to policy and standards. In commercial product development, systems engineering security concerns are driven by market acceptance, measured in continued product-line revenues and competitive position. Hybrid projects include commercial acquisitions of one-off systems (such as custom banking systems) and government projects of repetitive system deployments (like Predator drones).

Security technology has relied principally on reverse engineering successful breeches after the fact, and developing ways to recognize and prevent those attempts when tried again. This is true in the evolution of cyber security with firewalls and intrusion detection mechanisms, in physical security with various intrusion sensors and anti-tamper devices, in human security with security-clearance methodologies and employee behavior profiling, and now moving into supply-chain security with trusted supplier qualification and commercial off-the-shelf (COTS) device and software security testing.

As systems engineers we rely on the systems-security-engineering profession to know the methods of security breeches and to know the methods for detection and prevention. But after-the-fact analysis and countermeasure design are proving insufficient. The adversarial community innovates and evolves faster than breech analysis and countermeasure installation. Something different needs to be done, and systems engineering must enable the security engineer to do something different.

The Logic of SE Responsibility for Systems Security

Systems are engineered with expectations: to provide services or carry out missions that justify the development, production, and sustaining of investments. The return on investment occurs over time. Usually a period of many years is required. Value fails to accrue if the system's life or its ability to carry out its mission during that life is less than required. System lifetime, protection of critical system information, and critical assets that may be protected by a system, are under threat by competitive entities, as well as by unanticipated situations. System security is the property that guards against and counters these threats—a purposefully engineered property that can only emerge successfully from thoughtful system engineering.

Emerging technology is a double-edged sword. Modern technology is both the enabler of remarkable system capability and a source of constantly-evolving adversarial attack. The increasing use, knowledge, and complexity of digital data, control systems, and communication networks compel both new system capability and new vectors for system compromise. Accessibility to technologies such as global positioning systems, drones, and biological intervention bring new capability to physical system intervention. Globalization and outsourcing have made supply-chain insertion a successful new vector of system intervention. Moreover, enduring human factors of selfish interests, ideological motivation, and occasional faculty impairment make the insider threat always likely and multidimensional.

Within the systems engineering taxonomy, security is classified as a specialty engineering activity. To be sure, special knowledge, experience, and practice are necessary in system security engineering; especially when systems of all kinds are targets for intelligent, resourceful adversaries intent on system compromise. Security engineering is engaged to make a system secure, but when allocated solely to a separate specialty activity, this engagement is constrained by the nature of an already defined and often implemented system, or limited to ensuring that called-for standards and regulations are met. Constrained evolution of existing systems, and characterization as a compliance activity, hamstring the ability of security engineering to accept and dispatch system security responsibility effectively.

Dispatching Responsibility

Systems engineering is described and practiced as a collection of technical and project processes, organized for disciplined execution, with checks and balances throughout—in prudent practice. At the highest level the technical process of verification and validation, with test and evaluation, is focused on verifying that the system meets requirements and that the requirements are valid for meeting the system intent. As outlined in the INCOSE *Systems Engineering Handbook*, within each of the system engineering processes there are also formal internal checks and balances, called out to ensure the integrity of each process discipline.

Verifying and validating sustainable security of a system reaches back to the earliest two system engineering processes of defining stakeholder requirements and analyzing requirements, where requirements and the concept of operations govern what will be verified and validated for system security. Important outputs of the requirements analysis relevant for system security include measures of performance, systems functions, and verification criteria. Systems functionality should not ignore those functions that are intended to provide sustainable system security, nor can dedicated system security functions preclude the need for all other functions to include appropriate internally-integrated security measures. The expertise for integrating sustainable security in the processes of stakeholder-requirements definition and requirements analysis is best provided by the specialty engineering resources of security engineering as full peers, enabling the rapid upgrade and augmentation of security measures.

The concept of operations should recognize the reality of an evolving and innovative threat environment. This recognition should influence system-architecture considerations that will facilitate sustainable system security measures, so that these measures can evolve continuously throughout development and throughout operational life.

System architecture enables or impedes system security, and is an early design activity where engagement of security engineering is important. System adversaries learn system-protective measures and change methods rapidly. Architecture must accommodate protective measures that can change just as rapidly, and resilience that can deliver functionality while under attack. These needs argue for a security architecture that is composed of loosely coupled encapsulated functional components that can be replaced, augmented with additional functionality, and reconfigured for different interconnections. Long system life expectancies are critically vulnerable to non-agile architectures.

In each of the system engineering technical processes, disciplined checks and balances are included to ensure process integrity. Each of these processes enable or constrain the end capability of sustainable system security; and thus warrants explicit attention and collaboration with the expertise of actively engaged security engineering resources.

Trade-off evaluation and decision are important functions of system engineering, but these evaluations and decisions should be informed and advised by the expertise of competent and thoughtful security-engineering resources. Competence is rooted in the depth of specialty knowledge, whereas thoughtfulness is enabled by the breadth of the full system's requirements and intent knowledge—which can only be obtained when security engineering is in full participation throughout all of the systems engineering processes.

Experience Speaks

The articles in this issue of INSIGHT are intended to help lower and remove the barriers to mutually effective engagement of systems and security engineers. The barriers are those perceived by systems engineers, security engineers, project managers, and program managers. Many of these articles provide experience examples that can help systems engineering accept and dispatch responsibility for the sustainable security of systems. Systems engineers must recognize that systems security cannot be effective if it is not integrated intimately with the system requirements, the concept of operations, the architecture, and all the other systems engineering processes through operation and disposal.

Management Initiatives to Integrate Systems and Security Engineering

From Raytheon, Lori Masso and Beth Wilson share a management initiative that places system security responsibility within the systems engineering processes. This responsibility is backed up with system engineering training that provides fundamental understanding of system security concepts and policies and addresses how to identify security requirements. It also provides enough knowledge of the security fields to be able to ask the right questions and know if the answer represents a reasonable approach. Lori is a Principal Systems Engineer at Raytheon Company, with seven years of experience in system security engineering. Beth Wilson is an INCOSE ESEP, INCOSE Systems Security Working Group cochair, a Principal Engineering Fellow at Raytheon Company, and US National Defense Industrial Association Development Test and Evaluation Committee cochair, with a PhD in electrical engineering from the University of Rhode Island (US).

Information Security: Shaping or Impeding Systems in the Future?

Ken Kepchar, ESEP, retired Chief Systems Engineer of the US Federal Aviation Administration in the NextGen office, and owner of EagleView Associates, offers systems engineering consulting and training with a focus on transportation-related issues. Ken's article raises concern over the landscape of shifting digital technologies that influence systems engineering decision making. He notes that new risks are being introduced while traditional system development efforts defer or ignore security considerations until after the functional architecture has been established. He outlines some commonly held security "myths" that need to be purged, some principles to employ for effective security integration, and adjustments to include security capabilities as contributing feature in system design.

What Does a Systems-Security Engineer Do and Why Do Systems Engineers Care?

Janet Oren suggests that integration of systems-security engineering with all systems engineering processes is on the cusp of achievement. She attributes this to growing expertise in the security engineering community, and to a more detailed process approach expected in 2013 from the US National Institute of Standards and Technology as Special Publication 800-160, *Systems Security Engineering*. Janet is a Technical Director and Systems Security Engineer for the US Department of Defense, with a PhD in systems engineering from Stevens Institute of Technology (Hoboken, US-NJ). She feels that the success of this integration will result in systems that protect information and are more resilient.

Addressing Attack Vectors Within the Acquisition Supply Chain and the Development Lifecycle

John Miller of The MITRE Corporation opens a discussion of supply-chain threat. From a systems engineering view he focuses on understanding and addressing the "attack vectors" used to exploit vulnerabilities in the system-acquisition supply chain and the system-development lifecycle, examining the intersection of attack vectors with activities of systems engineering. John is a systems engineer at the MITRE with expertise in system security engineering, software engineering and development, hardware–software integration, and project management. He is currently developing program-protection methodologies and frameworks for the US defense department's major acquisition programs.

Requirements Challenges in Addressing Malicious Supply-Chain Threats

Paul Popick and Melinda Reed continue the discussion of supply-chain threats with latest US Department of Defense state of practice for incorporating trusted system and network security requirements into the specifications for large, complex systems. They describe the current environment, the trends that are influencing the need for system security engineering, and the types of system security requirements and analysis techniques. Paul is a retired IBM Global Services Director of delivery excellence, cochair of the INCOSE System Security Engineering Working Group, and maintains a continuing interest in systems engineering and program management through teaching and consulting. Melinda Reed is the Deputy Director for Program Protection within the Deputy Assistant Secretary of Defense Systems Engineering organization of the office of the US Secretary of Defense.

Uncertainty in Security: Using Systems Engineering Approaches for Robust System Security Requirements

From Sandia National Laboratories (Albuquerque, US-NM), Ruth Duggan and Mark Snell address the complicating factors in developing system security requirements; suggesting that an expert system security engineer can help the systems engineer navigate these complications so that the resulting system will be robust against future threats and technical advances. Ruth is a Senior Member for the Institute of Nuclear Materials Management and on its Executive Committee, and works for Sandia as a Systems Analyst of Nuclear Security Systems. Mark is a Distinguished Member of the Technical Staff in the area of physical protection at Sandia.

Enabling Sustainable Agile Security Through Systems Engineering

Rick Dove notes that long-life systems will have functional upgrades and component replacements throughout their life. Continuous evolution of system security is necessary to maintain parity with a continuously evolving threat environment. He reviews agile architecture fundamentals that enable effective security evolution, the important role played by the concepts of operations, principles for fleshing out the architecture, and a framework for developing responsive requirements. Rick teaches agile and self-organizing systems at Stevens Institute of Technology, chairs the INCOSE working groups for Systems Security Engineering and for Agile Systems and Systems Engineering, and is CEO and principle investigator for security-technology contracts at Paradigm Shift International.

Security Engineering Models

From Sotera Defense Solutions, Bob Marchant integrates the systems engineering lifecycle model with the US National Institute of Standards and Technology Risk Management Framework used as a security engineering lifecycle model. He then walks through the activities and guidelines used in process models and system baseline models that structure the systems security engineering effort. Bob is a CISSP (Certified Information Systems Security Professional) and an ISSEP (Information Systems Security Engineering Professional), and a technical fellow at Sotera, with 35 years of systems engineering experience that includes 20 years involved with information-systems security.

Evaluation of Security Risks using Mission Threads

From the Software Engineering Institute, Carol Woody describes the use of mission thread security analysis as a tool for systems engineers to evaluate the sufficiency of software security requirements. She then shows the value and use of this approach with a detailed example of the Commercial Mobile Alert System, a system that disseminates geographically targeted emergency-alert notifications. Dr. Woody leads a research team at the Software Engineering Institute focused on cyber security engineering: building capabilities in defining, acquiring, developing, measuring, managing, and sustaining secure software for highly complex, networked, software-intensive systems.

System Integration at the Security Interfaces

Kevin Stoffell, a Cyber Security Architect with the Battelle Memorial Institute, notes that security in information-technology systems is typically distributed, with many components relying on other components to ensure some or all of their security. Kevin suggests that this distributed interdependency poses some problems with integration process. He provides an example to illuminate the nature of the problems, and suggests that systems engineering interface control documents can and should be used as support to overcome these problems in the system certification, accreditation and authorization processes.

Verifying Security Control Requirements and Validating their Effectiveness

From Thales Australia, Bruce Hunter illuminates the planning and methods for system security verification and validation, and addresses continued verification and validation throughout a system's operational lifetime. He stresses the need for an adaptable approach that accommodates new emerging or discovered threats and vulnerabilities. Notably, he advises setting the scope of security testing beyond the identified system security requirements, to include any path that a threat may exploit. Bruce works in quality, security, and safety assurance for Thales, and holds CISM (Certified Information Security Manager) and CISA (Certified Information Systems Auditor) credentials from the Information Systems Audit and Control Association.

An Approach to Integrate Security into a Systems Engineering Curriculum

Don Gelosh wraps up our theme by addressing the education of systems engineers, proposing that system security consciousness and knowledge be integrated throughout the curriculum, especially in courses that deal with requirements, architecture and design, risk management, integration and test, sustainability, scalability and flexibility. He provides a framework for consideration and tailoring by institutions offering degrees in systems engineering, and speaks from personal experience and responsibility. Don is a CSEP-Acq, the first Director of Systems Engineering Programs at Worcester Polytechnic Institute, and is Level III qualified in the US Department of Defense systems planning, research, development, and engineering career field.

A Closing Thought

We cannot put security and system sustainability (an ility in name only?) into the functional category, as that category has historical meaning that refers directly to system functional requirements of delivered features. But that seems fuzzy. Security will not have the priority it needs until it is recognized as a functional requirement. Note that an insecure system is inherently "non-functional." Is this all a semantic game, or is it a game of I-Don't-Want-To-Have-To-Think-About-That?