

ENGINEERING FOR SYSTEMS ASSURANCE – A STATE OF THE PRACTICE REPORT

Paul R. Croll
CSC
5166 Potomac Drive
King George, VA 22485
+1 540 644 6224
pcroll@csc.com

Abstract - Those who acquire, build, and manage large-scale systems and Systems of Systems, recognize the complex supply chain they represent, consisting of proprietary and open-source software, legacy systems, hardware, and firmware; from multiple suppliers who employ people from around the world. As a result, the threat to today's systems is present across the full system life cycle. Dealing with that threat in the acquisition, development, operation, and maintenance of systems is largely a question of understanding and accepting residual risk, that is, the risk that still remains after all mitigation efforts has been employed. In this context, system assurance can be viewed as the level of confidence that the system functions as intended and is free of exploitable vulnerabilities, either intentionally or unintentionally designed or inserted as part of the system. Engineering practices that support such confidence in all phases of the life cycle are key factors to marketplace acceptance, corporate risk exposure, and to national security.

This paper addresses joint industry and Government efforts to understand the strengths and weaknesses of current engineering practices with respect to system assurance, and to provide recommendations for improvement. It covers the definition of the problem, results from several joint Industry/Government Forums addressing issues in systems assurance, collaboration efforts with industry consortia and standards bodies, and the current guidance for system acquirers, developers, operators, and maintainers.

THE ASSURANCE PROBLEM

In the cover letter to its 2005 report entitled *Cyber Security: A Crisis of Prioritization* [1], the President's Information Technology Advisory Committee (PITAC) captured the essence of what has been recognized as a growing systems assurance problem:

The IT infrastructure is highly vulnerable to premeditated attacks with potentially catastrophic effects. Thus, it is a prime target for cyber terrorism as well as criminal acts. The IT infrastructure encompasses not only the best-known uses of the public Internet – e-commerce, communication, and Web services – but also the less visible systems and connections of the Nation's critical infrastructures such as power grids, air traffic control systems, financial systems, and military and intelligence systems. The growing dependence of these critical infrastructures on the IT infrastructure means that the former cannot be secure if the latter is not.

The PITAC also noted in a 1999 report [2] that our technologies to build reliable and secure software are inadequate, explaining that our ability to develop software has not kept pace with hardware advances, and that we are also challenged in our ability to construct complex software-intensive systems for which we can anticipate performance.

Why the focus on software issues in *systems assurance*? Most systems we encounter today contain software elements and most depend upon software for a good portion of their functionality.

In the United States, industry and Government have been collaborating on understanding the issues associated with systems assurance, identifying the strengths and weaknesses of our current engineering practices, and providing guidance for improvement. In 2004, the U.S. Department of Defense (DOD), in cooperation with the U.S. Department of Homeland Security (DHS) began a series of Software Assurance Forums, to collaboratively address these problems. Early findings [3] included:

- Mission risk has dramatically increased due to the simultaneous growth in software vulnerabilities and in threat opportunities
- Risk management processes inadequately address these threats and risks
- Threats presented by suppliers of software products and services are not adequately identified and analyzed
- Development and acquisition processes inadequately address software security thereby contributing to the introduction of software vulnerabilities
- There is a fundamental lack of both the scientific understanding of software risks and the capabilities to effectively diagnose and mitigate in the in a timely manner

In 2005, the National Defense Industrial Association (NDIA) convened a Software Assurance Summit [4] to bring together Government and Industry in partnership to explore the range of opportunities for a long term solution to the issue of software assurance. The issues addressed included the following:

- There has been a dramatic increase in the importance of software in the functionality of defense systems, including both weapons systems and command/control and information systems.
- The software content of such systems is most often an amalgamation and integration of various software subsystems, using both new and legacy content, from a myriad of sources.
- It is often difficult to assure the source of software, from a supply chain perspective, given that software development often spans multiple companies and countries.

- Fully guaranteeing the integrity of systems from an information and software assurance perspective has become increasingly difficult.

In 2006, the NDIA convened a Top Software Issues Workshop [5] to examine the current most critical issues in software engineering that impact the acquisition and successful deployment of software-intensive systems.

The workshop identified eighty-five issues for further discussion, which were consolidated into a list of the top seven. Of those seven issues impacting software-intensive systems throughout the life cycle, two emerged that were focused on specifically on systems assurance

The first systems assurance-specific issue was:

There is a failure to assure correct, predictable, safe, secure execution of complex software in distributed environments.

Like the NDIA Software Assurance Summit preceding it, the Top Software Issues Workshop participants indicated that contributing to this issue was the supply chain pedigree of components and the adequacy of techniques for specifying, building, demonstrating, and verifying assured components.

From the threat perspective, software is inherently vulnerable to widespread assurance threats that can pose significant performance risks with extreme consequences. As more and more systems integrate software components of unknown pedigree, it is becoming increasingly difficult to determine their resistance to threats from adversaries in a globally interconnected environment.

From the engineering perspective, current techniques are inadequate for specifying, building, demonstrating, and verifying assured components with well understood properties, at least in a cost-effective and scalable manner. For example, one cannot easily infer the assurance properties of a system, or systems of systems, from component level assurance information. We just don't know enough about composability problems and emergent behavior when components are interconnected in large-scale systems and systems of systems. In addition, exhaustive testing to rule out vulnerabilities is generally not feasible due to the size and complexity of our systems of interest.

The second systems assurance-specific issue was:

Inadequate attention is given to the total lifecycle issues, including impacts on lifecycle cost and risk associated with the use of commercial or reused products and components.

Again, like the NDIA Software Assurance Summit preceding it, the Top Software Issues Workshop participants indicated that contributing to this issue was the lack of methods for accurately estimating life cycle costs when integrating commercial or reused products and components, the lack of attention to sustainment issues as well as issues related to open source licensing.

With respect to life cycle costs, the participants indicated that hidden costs and tradeoffs are not always fully considered when evaluating the appropriateness of using commercial or reused products and components instead of developing custom components. Cost analyses and tradeoffs must consider the impact of vulnerabilities, i.e. the pedigree of the code, when making make vs. buy decisions. Even where effective best practices for addressing life cycle costs issues are known, they are not consistently implemented.

With respect to sustainment, the participants indicated that there is insufficient attention paid to sustainment issues early in the life cycle, including licensing, and product support. This leads to a myriad of problems when commercial products or components inevitably change or when their suppliers either discontinue support or go out of business. In the hardware world Diminishing Manufacturing Sources and Materials Shortage (DMSMS) analyses are generally done when integrating commercial components as part of an approach for managing the risk of obsolescence [6]. DMSMS analyses focus on supplier viability, for the product of interest, and identification of alternate sources. This is especially important when considering trusted suppliers. The participants indicated that similar analyses should be done for software components.

With respect to open source licensing, the participants indicated that use of open source software can expose organizations to liability, loss of data rights, vulnerabilities, and potentially large amounts of rework.

ENGINEERING FOR SYSTEMS ASSURANCE

Stakeholder discussion over the last several years has demonstrated a reasonably consistent view of the problem space. System assurance, then, can be viewed as the level of confidence that the system functions as intended and is free of exploitable vulnerabilities, either intentionally or unintentionally designed or inserted as part of the system. The systems engineering challenge, with respect to assurance, is in integrating a heterogeneous set of globally engineered and supplied proprietary, open-source, and other software; hardware; and firmware; as well as legacy systems; to create well-engineered integrated, interoperable, and extendable systems whose security, safety, and other risks are acceptable – or at least tolerable.

The NDIA Top Software Issues Workshop [5] also addressed specific recommendations for action for both Government and Industry regarding the issues described above.

The recommended actions included:

- Collaborate to develop new approaches and improve existing approaches, standards, and tools that address systems assurance issues throughout the acquisition life cycle and the supply chain.
- Integrate systems and software engineering practices for producing system architectures and resulting systems that are resistant to intrusion and compromise.
- Sponsor research into new modalities for system composition to meet specific assurance objectives.
- Define software assurance quality attributes that can be addressed during architectural tradeoffs.
- Encourage the development of commercial standards addressing vulnerability management throughout the supply chain, including product-level and component-level specifications and standards for detecting component vulnerabilities.
- Develop policy, guidance, and training for the acquisition of systems with desired assurance properties.
- Improve and expand guidelines for addressing the total lifecycle issues associated with commercial and reused software.

- Ensure that life cycle issues and tradeoffs associated with the incorporation of commercial and reused software into systems are clearly addressed in program plans, and during periodic reviews.

Collaboration in developing guidelines for engineering practices for systems assurance is well underway. Efforts are ongoing through the NDIA System Assurance Committee, the DHS Software Assurance Forum, as well as in collaborative efforts between the Institute of Electrical and Electronics Engineers (IEEE) Software and Systems Engineering Standards Committee (S2ESC) and ISO/IEC JTC1/SC7, the ISO/IEC subcommittee on software and systems engineering standards.

NDIA Guidebook Effort

The NDIA in collaboration with DOD and other U.S. Federal agencies is developing a guidebook intended to supplement the knowledge of systems (and software) engineers who have responsibility for systems for which there are assurance concerns. The guidebook is entitled, "Systems Assurance – Delivering Mission Success in the Face of Developing Threats." [7] This guidebook is intended to:

- Provide practical guidance for Government, industry, and academia.
- Synthesize knowledge from foundation documents representing existing practice, policy, and guidance
- Present important concepts and principles from these source documents and discuss them in the larger context of systems assurance

The guidebook contains an introduction; a list of contacts in applicable communities of interest and practice; a section detailing correspondence with existing documentation and standards, including public law, government policy, and government and commercial standards; and a chapter describing systems assurance concepts and methodologies, including a discussion of the assurance case.

Section six of the guidebook is organized in accordance with the top level outline of ISO/IEC 15288, System Life Cycle Processes [8]. The intent of this organization was to provide an

established process framework for specifying additional systems assurance activities.

ISO/IEC 15288 itself is organized into Agreement Processes, Enterprise Processes, Project Processes, and Technical Processes.

The *Agreement Processes* facilitate interactions between the acquirer and the supplier. They establish the business relationship, including acceptance of the delivered system, and payment of the supplier upon its acceptance. They include the:

- Acquisition Process
- Supply Process

The *Enterprise Processes* facilitate management and improvement of the organization's business, including the management of its resources and assets and the risks it faces in the marketplace. These processes include the:

- Enterprise Environment Management Process
- Investment Management Process
- System Life Cycle Processes Management Process
- Resource Management Process
- Quality Management Process

The *Project Processes* facilitate project planning, management, and oversight. They address cost and schedule, achievement of project objectives, to oversight of project performance and corrective action when necessary. They include the:

- Project Planning Process
- Project Assessment Process
- Project Control Process
- Decision-making Process
- Risk Management Process
- Configuration Management Process
- Information Management Process.

The *Technical Processes* facilitate technical activities throughout the life cycle. They are the processes that create the system to satisfy the needs of the stakeholders; to operate, and maintain the system, throughout its life cycle, and eventually dispose of the system when its life cycle ends. These include the:

- Stakeholder Requirements Definition Process
- Requirements Analysis Process

- Architectural Design Process
- Implementation Process
- Integration Process
- Verification Process
- Transition Process
- Validation Process
- Operation Process
- Maintenance Process
- Disposal Process

The guidebook is a work in progress. It is the intent of the guidebook team to provide meaningful guidance on the systems assurance implications for each of the processes listed above, as well as case examples and documentation templates where appropriate.

Other DOD System Assurance Efforts

In addition to collaboration with industry through the NDIA, and with other Government agencies, the U.S. DOD is addressing the systems assurance problem both organizationally and through the development of additional guidance.

In 2006, the Deputy Undersecretary of Defense for Acquisition and Technology (DUSD/A&T) established a Deputy Director for Software and System Assurance under the Director, Systems and Software Engineering. This organization is moving forward in the following areas by [9]:

- Validating the need for systems assurance
- Defining the DOD strategy and concept of operations
- Providing a framework for integration of policies and guidance for acquisition of assured systems
- Integrating systems assurance into the acquisition lifecycle
- Providing coherent guidance for program managers
- Providing a focal point to endorse systems assurance, facilitate issue resolution, and advocate for Program Managers.

The DOD has described a vision of success for its systems assurance efforts. It looks like this:

- The requirement for assurance is allocated among the right systems and their critical components
- DOD understands its supply chain risks
- DOD systems are designed and sustained at a known level of assurance

- The commercial sector shares ownership and builds assured products
- Technology investment transforms the ability to detect and mitigate system vulnerabilities

The organizational infrastructure is now in place to begin to make this vision real.

In addition, the DOD Information Assurance Technical Assistance Center (IATAC) and the DOD Data and Analysis Center for Software (DACS) have jointly authored a *State of the Art Report on Software Security Assurance* [10] that is in its early draft stage. The purpose of this report is to identify and describe the current “state of the art” in software security assurance, including trends in the following areas:

- Techniques for the production of secure software
- Technologies that exist or are emerging to address the software security challenge
- Current activities and organizations in government, industry, and academia, in the U.S. and abroad, that are devoted to systematic improvement of software security
- Research trends worldwide might improve the state of the art for software security

DHS Software Assurance Program

In addition to the stakeholder discussions resulting from the DHS/DOD Software Assurance Forums, several products are in development. They include: *Secure Software Assurance: A Guide to the Common Body of Knowledge to Produce, Acquire, and Sustain Secure Software* [11]; *Security in the Software Life Cycle: Making Software Development Processes – and the Software Produced by Them – More Secure* [12]; and *Software Assurance in Acquisition: Mitigating Risks to the Enterprise* [13].

Secure Software Assurance: A Guide to the Common Body of Knowledge to Produce, Acquire, and Sustain Secure Software is intended as a framework to identify workforce needs for competencies and leverage standards and best practices to guide software-related curriculum development. It addresses three domains: acquisition and supply, development, and post-release assurance.

Security in the Software Life Cycle: Making Software Development Processes – and the

Software Produced by Them – More Secure is intended to be a compendium of methodologies, life cycle process models, sound practices, and supporting technologies that would, if adhered to, increase software security.

Software Assurance in Acquisition: Mitigating Risks to the Enterprise is intended to provide guidance on enhancing supply chain management through improved risk mitigation and contracting for secure software.

These reports are just a small part of DHS efforts to provide an enabling foundation for assurance and trustworthiness [14]. The overall DHS software assurance effort addresses three primary areas:

- Trustworthiness
- Predictable Execution
- Conformance

Trustworthiness entails ensuring that no exploitable vulnerabilities exist, either maliciously or unintentionally inserted.

Predictable Execution entails providing justifiable confidence that software, when executed, functions as intended

Conformance entails a planned and systematic set of multi-disciplinary activities ensures software processes and products conform to requirements, standards/ procedures.

Standardization for Systems and Software Assurance

In addition to the standards produced for secure coding by ISO/IEC SC22, (Languages); for IT Security by ISO/IEC SC27 (IT Security Techniques), and for safety by IEC SC 65A (Functional Safety); ISO/IEC JTC1/SC7 (Software and Systems Engineering), and the IEEE Computer Society, have undertaken an effort to produce a standard for Systems and Software Assurance. This standard will provide requirements for the development, operation, and maintenance of systems and software products that are required to exhibit properties related to safety, security, and dependability. It will do this by providing requirements for processes, activities, and tasks that are in addition to those of ISO/IEC 15288 and ISO/IEC 12207, Software Life Cycle Processes [15]. In addition, it will also

address requirements for information artifacts that result from those processes that would be in addition to those of ISO/IEC 15289, Guidelines for the Content of Software Life Cycle Process Information Products (Documentation) [16].

SUMMARY

The systems assurance problem we face today has been clearly identified. The systems engineering challenge, with respect to assurance, is in integrating a heterogeneous set of globally engineered and supplied proprietary, open-source, and other software; hardware; and firmware; as well as legacy systems; to create well-engineered integrated, interoperable, and extendable systems whose security, safety, and other risks are acceptable – or at least tolerable.

The preceding discussion focused largely on software issues. Why address software issues in *systems assurance*? Most systems we encounter today contain software elements and most depend upon software for a good portion of their functionality

Joint industry and Government efforts are ongoing to understand the strengths and weaknesses of current engineering practices with respect to systems assurance. National and international standards efforts are also capturing and codifying minimum acceptable practice regarding engineering for systems assurance. These efforts are producing guidance regarding engineering practices that can impact the systems assurance challenge.

REFERENCES

- [1] President's Information Technology Advisory Committee (PITAC), *Cyber Security: A Crisis of Prioritization*. National Coordination Office for Information Technology Research and Development, Arlington, VA, 2005.
- [2] President's Information Technology Advisory Committee (PITAC), *Information Technology Research: Investing in Our Future*. National Coordination Office for Information Technology Research and Development, Arlington, VA, 1999.
- [3] J. Jarzombek. *DOD Software Assurance Initiative: Mitigating Risks Attributable to Software*. DOD Software Assurance Forum, July 2004.
- [4] K. Baldwin, *Welcome to the Software Assurance (SwA) Workshop & Summit*. NDIA Software Assurance Summit, National Defense

Industrial Association, Arlington, VA, September 2005.

[5] G. Draper (ed.), *Top Software Engineering Issues Within Department of Defense and Defense Industry*. National Defense Industrial Association, Arlington, VA, August 2006.

[6] *Diminishing Manufacturing Sources and Material Shortages (DMSMS) Guidebook*. Office of the Under Secretary of Defense Acquisition, Technology, & Logistics, November 2006.

[7] C. Powell and D. Kleiner (eds.), *Systems Assurance – Delivering Mission Success in the Face of Developing Threats*. National Defense Industrial Association, Arlington, VA, January 2007.

[8] ISO/IEC 15288:2002, *Systems Engineering — System Life Cycle Processes*, ISO/IEC JTC1/SC7, 2002. ISO, Geneva Switzerland, 2002.

[9] K. Baldwin. *DOD Software Engineering and System Assurance New Organization – New Vision*, DHS/DOD Software Assurance Forum, March 8, 2007.

[10] K. Goertzel (ed.), *State of the Art Report on Software Security Assurance, Draft*. DOD Information Assurance Technical Assistance Center (IATAC) and the DOD Data and Analysis Center for Software (DACs), March 2007.

[11] S. Redwine (ed.), *Secure Software Assurance: A Guide to the Common Body of Knowledge to Produce, Acquire, and Sustain Secure Software, Draft 1.1*. U.S. Department of Homeland Security, September 25 2006.

[12] K. Goertzel (ed.), *Security in the Software Life Cycle: Making Software Development Processes – and the Software Produced by Them – More Secure, Draft 1.1*. U.S. Department of Homeland Security, July 2006.

[13] *Software Assurance in Acquisition: Mitigating Risks to the Enterprise, Draft 1.0*. U.S. Department of Homeland Security, March 2007

[14] J. Jarzombek. *Software Assurance: A Strategic Initiative of the U.S. Department of Homeland Security to Promote Integrity, Security, and Reliability in Software*, IEEE Software and Systems Engineering Standards Committee, Executive Committee Winter Plenary Meeting, February 2007.

[15] ISO/IEC 12207:1995, *Standard for Information Technology — Software life cycle processes*. ISO, Geneva Switzerland, 1995.

[16] ISO/IEC 15289 FDIS 2006, *Software Engineering — Software Life Cycle Process — guidelines for the content of software life cycle process information products (documentation)*. ISO, Geneva Switzerland, 2006.