Editors: Vijay Varadharajan, vijay@ics.mq.edu.au
Fred Cohen, fc@all.net

Systems Security Engineering

Jennifer L. Bayuk
Stevens Institute of Technology

Systems engineers solve large problems by breaking them down into well-defined pieces, while preserving the problem definition for use in validating the solution. Traditional systems-engineering approaches have not until recently been applied to systems security problems. Systems security engineers have instead focused on security standards, so standards compliance now serves as evidence that security problems are addressed. A new systems-engineering security road map recommends that systems engineers and security engineers converge on empirical methods.

Security Standards versus Requirements

Recent data breaches and industrial-control-system incidents call attention to the inadequacy of current systems security approaches.[1,2] Each case presents more compelling evidence of cybersecurity threats' potential economic impact. Each case reinforces that we can't assume that the existing standards for technology control provide security. A vast amount of money has been directed toward cybersecurity solutions.[3,4] Yet, there is no proven method of deciding on what that money should be spent, and no new paradigms have evolved to guide management decisions toward practical security solutions.[5,6]

To date, cybersecurity curricula have concentrated mostly on the technology issues involved in implementing security standards. They haven't focused on any other means to measure systems security. Although one textbook attempted to model enterprise security using the Zachman enterprise architecture framework,[7] that attempt didn't result in any comprehensive way to model or measure the security of any given system. Even textbooks that combine security and engineering principles emphasize a security engineer mindset rather than suggest any innovative methods, tools, and procedures with which to approach systems security engineering.[8,9] Consequently, many security engineers use checklists to ensure their work is complete rather than validate that they've addressed systemic security requirements, whereas most security engineers don't have the systems-engineering background required to approach a security problem holistically.

Holistic system views of verification and validation are the systems engineer's forte.[10] However, when it comes to cybersecurity, systems engineers typically cede the responsibility to the security profession. As Barry Horowitz put it in a discussion session at the 2010 Systems Engineering Research Center (SERC) Security Workshop,

> When you ask an engineer to make your boat go faster, you get the trade-space. You can get a bigger engine but give up some space in the bunk next to the engine room. You can change the hull shape, but that will affect your draw. You can give up some weight, but that will affect your stability. When you ask an engineer to make your system more secure, they pull out a pad and pencil and start making lists of bolt-on technology, then they tell you how much it is going to cost.

One reason this situation is prevalent is that systems engineers haven't considered it a problem. They're taught to divide system requirements into two partitions: functional and nonfunctional requirements, or capabilities and characteristics. Capabilities always take precedence over characteristics, and security is classified as a characteristic.[11] One otherwise scholarly and astute textbook on systems engineering refers to security as "related to system attributes that enable it to comply with regulations and standards."[12] Blaming security standards bodies for a poor security design's outcome is easier than taking responsibility for "building security in." (This phrase, owing largely to the community behind the site https://buildsecurityin.us-cert.gov/bsi/home.html, has become

1540-7993/11/$26.00 © 2011 IEEE

synonymous with software security. However, I use it to refer holistically to any system of interest.) Although the systems-engineering literature has directly addressed security, it has circularly defined security as a process by which to ensure security concerns are covered, rather than as a core system requirement.[13,14]

As security practitioners search for workable solutions to the increasingly complex maze of malware they encounter, the trend should be to escape from best-practices checklists and return to core systems-engineering methods, processes, and tools. However, as I noted, most security engineers have no experience with these methodologies, which anyhow have traditionally obscured security requirements. As long as systems engineers don't consider security a functional requirement, it won't likely rise to the top of the implementation checklist. This is because processes for managing system development life cycles prioritize functional requirements over nonfunctional requirements. Security practitioners aren't getting help at the design stage, and we'll need new approaches to systems engineering to meet the growing need for secure systems.

One outspoken security practitioner, Ed Amoroso, discussed some of these issues in his book *Cyber Attacks*.[15] He presented several new ways of thinking about security that might offer a clue to security engineering's future. For example, he posed these questions:

- How might your system employ intentional deception?
- How might it diversify its threat surface?
- How might it increase situational awareness?

Where requirements-gathering techniques such as these are employed in the service of security, they present functional requirements at the systems level.

Security research, on the other hand, examines the existing practice of security as if it was an organic object of interest. Countless papers observe existing systems and networks and try to make sense of their security properties. Security practitioners frequently compare security researchers to alchemists searching for the formula for gold. The practitioner perception is that security research rarely uses scientific methods and that its activities appear faith based rather than fact based. I make this observation to contrast a promising emerging trend in security research with its checkered past.

A Fresh Look at Systems Security

This article builds on a foundation created in the SERC's *Systems Security Engineering: A Research Roadmap*.[16] The road map emphasized that progress in systems security research must follow a scientific process that includes clear problem statements, thorough problem background descriptions including a full literature review, clearly defined solution criteria, and proposed hypotheses formulated to shed light on a solution and how it can be proven or disproven.

The road map acknowledges that simply challenging the systems engineer to put aside security standards and start afresh won't resolve systemic security problems. The existing standards came about because security is a difficult problem. A generation of practitioners who entered the systems security field over the past 40 years embraced the current standards and models because they found common solutions to diverse security problems and shared them. This work is significant; we should leverage it by integrating it with a fresh look at the systems engineer's mission with respect to security.

The fresh look should start with a concept of security that allows it to be recognized as a tangible systems attribute. Security provides safeguards that contribute to a system's ability to achieve its mission and purpose in the face of changing threats. By this definition, requirements might include a security feature as a system capability with a clear, measurable goal, albeit one that's customized in context. A clear understanding of the definition of security in the context of a given system mission should let us design alternative security architectures, as well as metrics we can apply to those architectures to determine their effectiveness in maintaining system security.

Emerging security architecture frameworks following this methodology might extend and enhance systems architecture to produce security requirements at the system level rather than the security technology level.[17] The security metrics resulting from these efforts should play a key role in the development of new tools for systems security engineers. Of course, owing to the possibility of unknown threats, no system will ever be 100 percent secure. Nevertheless, this approach should enable system owners and operators to

The trend should be to escape from best-practices checklists and return to core systems-engineering methods, processes, and tools.

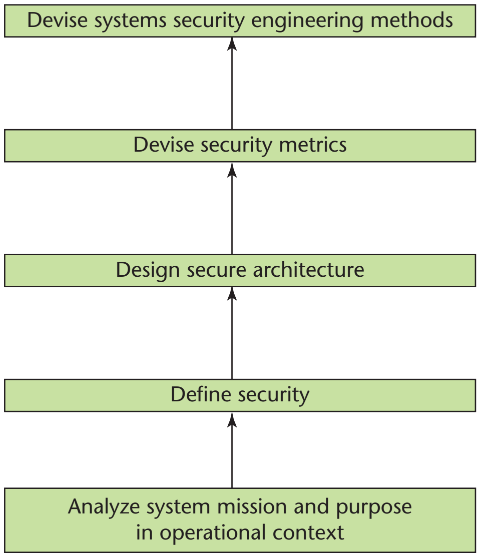
Devise systems security engineering methods
Devise security metrics
Design secure architecture
Define security
Analyze system mission and purpose in operational context

Figure 1. An approach to systems security engineering. Starting with the operational context, we define security in terms of the system mission. From there, we can develop architecture alternatives and both verification and validation criteria, which in turn form the basis for security metrics.

- identify security features requiring system-level functions,
- evaluate the extent to which security features protect systems from deliberate damage that would cause system failure, and
- devise verification and validation metrics at the system level that show that security requirements are met.

Figure 1 illustrates the approach. By extracting the definition of security from the context in which a system operates, we can integrate security architecture into systems architecture, customizing it rather than bolting it on. We can devise architecture metrics that measure whether security features meet security functional requirements. When systems exhibit similar architecture patterns, they may use common security architecture models. The existence of such models should let us develop tools to guide future engineering efforts toward more secure solutions.

A key element of *Systems Security Engineering: A Research Roadmap* is to enable security researchers to assess the value of their potential contribution to the field. An engineering approach to verification and validation of security requirements will ipso facto provide a methodology to test a research hypothesis. Security research employing such methods should be able to build on prior results by citing successful verification and validation results in similar architecture patterns. Such an engineering approach might seem like practical application rather than research to some in our community. However, they might be overlooking that security isn't at all well understood, so any serious investigation of its properties constitutes a research endeavor. □

References

1. L. McGlasson, "More Heartland-Related Fraud Detected," *Bank Information Security*, 1 Apr. 2010; www.bankinfosecurity.com/articles.php?art_id=2366.
2. V. Fuhrmans, "Virus Attacks Siemens Plant-Control Systems," *Wall Street J.*, 22 July 2010; http://online.wsj.com/article/SB10001424052748703954804575381372165249074.html.
3. C. Williams, "Cameron to Spend £1bn+ on Cyber Security," *Register*, 14 Oct. 2010; www.theregister.co.uk/2010/10/14/cyber_budget.
4. M. Cacas, "DHS Outlines Cybersecurity Planning," Federal News Radio, 21 July 2010; www.federalnewsradio.com/?nid=35&sid=2007741.
5. E. Spafford, "Privacy and Security Remembrances of Things Pest," *Comm. ACM*, vol. 53, no. 8, 2010, pp. 35–37.
6. G. Shipley, "Outgunned: How Security Tech Is Failing Us," *Information Week*, 9 Oct. 2010; www.informationweek.com/news/security/antivirus/showArticle.jhtml?articleID=227700363&queryText=Greg%20Shipley.
7. J. Sherwood, A. Clark, and D. Lynas, *Enterprise Security Architecture*, CMP Books, 2005.
8. R. Anderson, *Security Engineering*, 2nd ed., John Wiley & Sons, 2008.
9. M. Bishop, *Computer Security: Art and Science*, Pearson Education, 2003.
10. P. Checkland, *Systems Thinking, Systems Practice*, John Wiley & Sons, 1999.
11. D.M. Buede, *The Engineering Design of Systems: Models and Methods*, John Wiley & Sons, 2009.
12. W. Larson et al., *Applied Space Systems Engineering*, McGraw Hill, 2009.
13. *Information Technology—Systems Security Engineering—Capability Maturity Model*, Int'l Org. for Standardization and Int'l Electrotechnical Commission, 2002.
14. *Systems and Software Engineering—Systems and Software Assurance—Part 2: Assurance Case*, Int'l Org. for Standardization and Int'l Electrotechnical Commission, 2009.
15. E. Amoroso, *Cyber Attacks*, Elsevier, 2010.
16. J. Bayuk et al., *Systems Security Engineering: A Research Roadmap*, tech. report SERC-2010-TR-005, Systems Eng. Research Center; 2010; www.sercuarc.org/fileadmin/sercuarc/SERC_Publications/SERC-2010-TR-005-Security-100823_01.pdf.
17. J.L. Bayuk and B.M. Horowitz, "An Architectural Systems Engineering Methodology for Addressing Cyber Security," to be published in *Systems Eng.*, vol. 14, no. 3, 2011.

Jennifer L. Bayuk *is the director of the Stevens Institute of Technology's Systems Security Engineering program. Contact her at jennifer@bayuk.com.*