

SoS configuration. An example visualisation is depicted in figure 4.

Conclusions

The implementation of the high-level security requirements deriving from data policies is critical to gaining participation of systems in an SoS by reassuring systems managers and owners about the use and dissemination of their data. This short article illustrates an overview of a model-based approach that we have introduced to support SoS security engineering for data policies. The approach includes the formal definition of a data-policy concept and intuitive methods for the derivation of high-level security requirements. The approach also includes the injection of these high-level security requirements in the definition of the SoS's functional architecture. Similarly, it is possible to verify that the physical SoS architecture meets the high-level security requirements. The approach is part of the European Space Agency Architectural Framework, thus providing an integrated means for security engineering within the entire SoS engineering process. Graphical and interactive visualisation tools are also provided to more effectively manage the design complexity of the architectural and security issues. 

References

- Bayuk, J.L. 2011. "Systems Security Engineering." *IEEE Security & Privacy* 9 (2): 72–74.
- Eisenmann, H., J. Miro, and H. P. De Koning. 2009. "MBSE for European Space-Systems Development." *INSIGHT* 12 (4): 47–53.
- ESA (European Space Agency). 2011a. Galileo: Navigation. <http://www.esa.int/esaNA/galileo.html>.
- _____. 2011b. European Space Situational Awareness. <http://www.esa.int/esaMI/SSA/index.html>.
- Gianni, D., J. Lewis-Bowen, N. Lindman, and J. Fuchs. 2010. "Modeling Methodologies in Support of Complex Systems of Systems Design and Integration: Example Applications." Paper presented at the Fourth International Workshop on System and Concurrent Engineering for Space Application, Geneva, Switzerland, 13–15 October.
- Gianni, D., N. Lindman, S. Moulin, and J. Fuchs. 2011. "SSA-DPM: A Model-based Methodology for the Definition and Verification of European Space Situational Awareness Data Policies." Paper presented at the European Space Surveillance Conference, Madrid, Spain, 7–9 June.
- GMES (Global Monitoring for Environment and Security). 2011. GMES info. <http://www.gmes.info>.
- Open Group. 2011. TOGAF 9TM Enterprise Edition.
- UK Ministry of Defense. 2009. MOD Architecture Framework (MODAF). <http://www.modaf.org.uk/>.
- VEGA. 2010. *SoSDEN Project Final Report*. European Space Agency.

Systems-of-Systems Issues in Security Engineering

Jennifer L. Bayuk, jennifer.bayuk@incose.org

A Systems Concern

Automated control systems play a large role in both cyber and physical infrastructure. As such systems service well-defined communities in a given nation-state, each system has inherent, or inherited, adversaries. Growing adversary awareness concerning the weaknesses in these systems has led to increases in both the number and variety of systems-security incidents. Publicly visible cyber events were in the past limited to disruptions of cyber operations, whether desktop or data-center. Cybersecurity adversaries are now disrupting operations in everything from power plants to cell phones. Nevertheless, this recognition appears to be limited to a small but increasing number of system-security professionals, and often hides in plain sight from mainstream systems engineers.

Many people are familiar with Stuxnet, the worm that disrupted operations at Iran's nuclear power plant. But few are familiar with the extent to which control systems have already been subject to tampering due to inadequate security controls (Weiss 2010). The degree of the threat should be made even more visible by reports of accidental system malfunctions that are inadequately addressed by control measures that seem like obvious electronic safeguards. For example, the United States National Transportation Safety Report for the San Bruno pipeline incident in California in September 2010 included the revelation that a temporary power loss in upstream equipment triggered an automated control to set a regulating valve to full-open mode, yet the investigators failed to recommend improvements in electronic control-system design, but concentrated their recommendations on the material composition of the pipeline itself.

State of the Practice

To date, security-engineering guidance has been concentrated on the technology issues involved in implementing security standards (Bayuk 2011). It has not focused on any other means to effect or to measure systems security. Although one textbook attempted to model enterprise security using the Zachman enterprise-architecture framework (Sherwood 2005), that attempt did not result in any comprehensive way to model or measure the security of any given system, and that publication is out of print. Even textbooks that combine security and engineering principals emphasize the mindset of the security engineer rather than suggest any innovative methods, tools, or procedures with which to approach systems-security engineering (Anderson 2008; Bishop 2003).

The result is that security engineers leave security requirements to security standards-setting bodies, and use security standards documents as checklists to ensure their work is complete rather than validate that systemic security requirements are addressed. 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” (Larson 2009, 114). As one colleague observed in a recent paper, “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” (Horowitz 2010).

This practice makes sense to security professionals simply because they have no other way of meeting their job-function requirements. A typical security engineer is a shared resource in a vast sea of projects, who will never understand the full set of objectives for most missions. In preparation, security engineers arm themselves with a security-standards document, a hammer that turns every project into a nail. They turn over all judgment on cost–benefit trade-offs to more informed “decisionmakers” who are not well-versed on security issues, and so often not qualified to make trade-space decisions on security controls. However, this leaves the security engineer off the hook for bad security-design choices, so that everyone makes choices they can justify and nobody is responsible for overall system security. The situation is exacerbated by the fact that intelligent design is rarely applied to the security-control alternative choices because the security-standards document has already supplied them.

The state of the security-engineering practice was recently the subject of a research program at the Systems Engineering Research Center (<http://www.searc.org/>). The goal of the project was to bring an empirical approach to the study of systems-

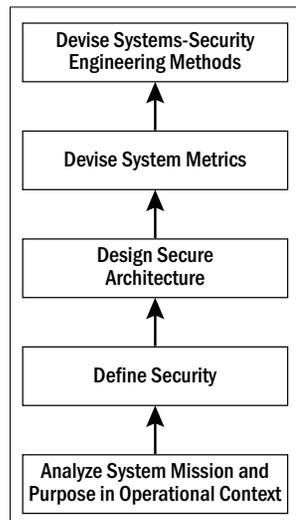


Figure 1. SERC approach to systems-security engineering

security engineering, and to establish a foundation upon which security research results may effectively address systemic security issues going forward. The output of the project was a roadmap for systems-security research (Bayuk 2010). The report emphasized that progress in system-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 a proposed hypothesis, formulated to shed light on a solution and how it may be proven or disproven. The idea is to challenge the assumption that anyone can in advance create a list of technology-control measures that will apply to every project, even all projects in a given domain. The general approach is illustrated in figure 1.

Issues Relating to Systems of Systems

Security issues in systems of systems include all the security issues faced by individual systems, and systems of systems also introduce issues related to transitivity and composition, the most obvious of which concerns trust. Trust is a recurring security issue in the literature of SoS security. Though no generally agreed-upon definition of trust exists in any systems context, the foundations of many SoS security-planning processes are littered with assumptions concerning trust models.

For example, consider the case of the National Health Information Network in the United States (see US Department of Homeland Security 2011). In this network, any health-care provider, including a doctor’s office, will be a node on a network with the ability to pull down any health-care records in a national virtual repository created by a system of cooperating systems. Yet the primary control at each network interface is a data-use and reciprocal-support agreement (DURSA), which is a legal document, as opposed to any data-level security features. The plan is to shift the burden of security engineering down to every doctor’s office rather than to address it at the SoS planning stages. Although this may be an extreme example, overreliance on trust models is rampant in the SoS projects and has been since the dawn of the Internet.

Even when SoS planning seems confined to a single enterprise, trust assumptions enable systems engineers to overlook security issues. As enterprises sequentially develop new systems and modernize existing systems that support various functions or various organizations, they rely on business partners and services to support various automated interfaces. They contractually obligate compliance with industry security standards rather than incorporating security as a design requirement for the emergent SoS. For example, data-security breaches in payment-card processors and credit-reporting companies allow banks to disclaim responsibility for privacy breaches. It is a common occurrence that both internal and external systems that were originally independent are integrated to achieve greater value, resulting in an SoS where security issues are pushed further and further away from the original system scope.

Even when both the SoS planning and the operations environment are confined to a single enterprise, trust assumptions act as security blinders. While the overall SoS serves the enterprise, each individual system lifecycle may be managed and operated by a different organization. Each organization manages its own risk-assessment process, and security requirements for data handling in one organization do not necessarily translate to another. This was the root cause of the Heartland Payment Systems security breach in January 2009, where corporate operations managed a network that was used to deploy management utilities into their credit-card network, but the security of that network did not meet the standards for credit-card networks, and it became the back door through which the hackers stole credit-card data (Mogull 2009).

Potential Solutions

As there is currently a lack of recognition that even standalone systems-security requirements may be unique, this issue must be confronted before SoS security issues will ever be adequately addressed. As previously mentioned, SERC has taken a first step in that direction. In the SERC security-engineering approach, the mission and context of an SoS would drive the definition of security for a constituent system, as well as a systems architecture that includes security features at both the individual system and interface level. For example, security may be defined generically as an attribute of the system that thwarts perpetrators who enact threats to exploit vulnerabilities that permit system disruption. To attribute this property to an SoS, the requirement must be met by a set of system features that are properly considered security features. These features would be part of the SoS architecture rather than an addition to it. They would address security requirements for each individual system, as well as minimize the impact to SoS operations due to systemic security threats and vulnerabilities introduced by composition. For example, design patterns can be developed where peer systems are used to help isolate the existence of a difficult-to-detect attack that manipulates or steals data. This can be accomplished, for example, through data parsing and continuity checking wherever data crosses the boundaries of an individual system and serves one of its peers. Design patterns such as these provide a starting point for exploring the flexibility of the practical management constraints that limit SoS solutions, so that over time the systems engineering community can establish a generally accepted understanding of what is deemed acceptable from an SoS security-management point of view, and what is not.

SoS security-architecture patterns are likely to include derivatives of related patterns used for a single system. For example, an SoS-security attack-detection pattern may be considered an extension of the each individual system's intrusion-detection capability in that the design in both cases should be supported by a peer-monitoring system in addition to its own processing; if its own monitoring is compromised, the peer system should alert. However, in the single-system case, the peer detection is usually passive and noninterfering, whereas in the SoS case it may be acceptable to suffer some losses in performance to gain the benefit of attack detection at the community level. That is, an automated intrusion capability may trigger an automated response to temporarily shut down an interface to the compromised system in an SoS community. Security metrics could be devised that would be useful in judging the security level of such a solution, for example, the percentage of peer interfaces supported by community attack-detection, the coverage in terms of network and data protocols, the time it takes to accomplish basic detection capabilities, and the extent to which the community systems can

automatically respond to protect their interfaces. These also may also be extended to systems-of-systems reporting and alerting mechanisms to be used in cases of identified data leakage.

Conclusion

The application of security standards does not make a system secure. Where security requirements are acknowledged to be system-specific, a systems engineer is allowed to go to the drawing board to see how they may best be accomplished. Today's plethora of security standards and best practices should be considered a source of potential control mechanisms that may be mixed and matched with other control mechanisms to achieve overall system-security objectives. From this type of mission-defined objective, a more specific set of requirements may be established. This approach is expected to lead directly to the development of new architectural security metrics. The results of these activities should be improved relevance of security guidance to the general systems engineering workforce. 

References

- Anderson, R. 2008. *Security Engineering*. 2nd ed. Indianapolis, IN (US): Wiley.
- Bayuk, J. 2011. "On the Horizon: Systems Security Engineering." *IEEE Security & Privacy* 9 (2): 72–74.
- Bayuk, J., D. Barnabe, J. Goodnight, D. Hamilton, B. Horowitz, C. Neuman, and S. Tarchalski. 2010. *Systems Security Engineering: A Research Roadmap; Final Technical Report*. Stevens Institute of Technology, Washington, DC (US): Systems Engineering Research Center (<http://www.sercuarc.org>).
- Bishop, M. 2003. *Computer Security: Art and Science*. Boston, MA (US): Addison-Wesley.
- Horowitz, B. M. 2010. "Frameworks to Guide Cyber Security Solution Application on a Systems Engineering Basis." Paper presented at Systems Engineering Research Center Security Workshop, Washington, DC (US), 31 March 2010.
- Larson, W., D. Kirkpatrick, J. J. Sellers, D. Thomas, and D. Verma. 2009. *Applied Space Systems Engineering*. Boston, MA (US): McGraw Hill.
- Mogull, R. 2009. "An Open Letter to Robert Carr, CEO of Heartland Payment Systems." Securosis (blog). 12 August 2009, <http://securosis.com/blog/an-open-letter-to-robert-carr-ceo-of-heartland-payment-systems/>.
- Sherwood, J., A. Clark, and D. Lynas. 2005. *Enterprise Security Architecture*. San Francisco, CA (US): CMP Books.
- US Department of Homeland Security. 2011. Homeland Security Information Network. Website of the Department of Homeland Security. http://www.dhs.gov/files/programs/gc_1156888108137.shtm.
- Weiss, J. 2010. *Protecting Industrial Control Systems from Electronic Threats*. New York, NY (US): Momentum Press.