

## Systems of Systems and Self-Organizing Security

Rick Dove, rick.dove@incose.org; and Jennifer Bayuk, jennifer.bayuk@incose.org

The frontier of systems engineering is confronting systems of systems and self-organizing systems, so that we can develop early conceptual and operational understandings. INCOSE's Systems Science and Complex Systems Working Groups face this frontier on conceptual grounds, while the Autonomous Systems Test and Evaluation and Anti-Terrorism International Working Groups face it on operational grounds. The Systems Security Engineering Working Group engages on at least three fronts: conceptual, operational, and architectural.

From concept to operations, theory meets practice, and practice informs and refines theory. Where frontier interest is in systems of systems or self-organizing systems, that learning-loop can be fastest where the stakes are highest. Perhaps nowhere else are the stakes higher than where systems become high value targets under attack by determined intelligent adversaries. The point is made to suggest that system security provides an accelerated learning arena for testing and vetting theories of systems of systems and self organizing systems.

System-adversarial communities are bound by shared knowledge and common focus: to cripple, own, or repurpose systems of all kinds, for a wide range of motivations. These adversarial communities are natural systems of systems, composed of intelligent agents that leverage shared knowledge for situational awareness, innovation, and evolution.

Figure 1 shows the state of affairs. The systems engineering community, augmented (but rarely assisted) by the security engineering community, fields systems that exhibit static security profiles. A dynamic, intelligence-driven adversarial community probes these systems until it finds or invents methods of successful attack.

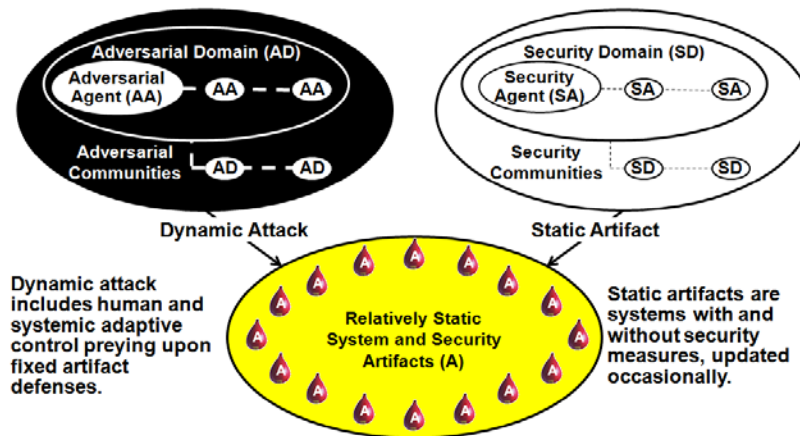


Figure 1. Yes, those are punching bags. The dotted lines in the adversarial and system circles represent the degree of collaborative security-innovation communication: fat for the adversary, almost nonexistent for systems-security communities.

Systems security has been likened to an arms race: an advance on one side encourages and fosters an advance on the other side. Unfortunately for systems security, it is the side of offense that leads the advances and the side of defense that tries to catch up but never jumps ahead. A successful catch-up reduces immediate losses but does not lead to a superior position, nor does it decrease the widening gap in security capability between attacker and defender.

The systems-security situation has also been called asymmetrical, citing the adversarial advantage of only needing to find a single vulnerability, while a successful defense needs to find

all vulnerabilities. Strategy based on eliminating vulnerabilities has not proven effective. On the contrary, some have argued that a better strategy would be to increase the pace of deployed system innovation and replacement, removing the adversary's long life learning curves and tool development advantages.

Resilient system strategies may be a more manageable way to counter the asymmetry of attack and defense. In recognition that systems will have vulnerabilities that adversaries will attack, and that system design needs mechanisms to weather successful attack and remain viable, engineers are now placing a new strategic priority on system resiliency. Survivability through resilient design is not a new concept, but still remains largely a research activity.

Another kind of asymmetry exists with less recognition: the systems-security situation is an ecology of predator and prey. Natural systems live with risk and have evolved mechanisms to continue species existence in the face of attack and loss. Natural predator-prey ecological systems do not generally maintain ecological balance by a symmetric turning of the tables. Rabbits have not evolved "tooth and claw," but rather maintain ecological balance and ensure survival of the species with a species resilience based on a tolerance for loss. Resilient system strategies can do the same, with high redundancy and rapid repair or replacement. For certain system types this approach has merit.

A third form of asymmetry exists that has little recognition but perhaps the most importance: adversarial communities lead in innovation and evolution, from developing improvised explosive devices to the recent Stuxnet weaponized malware effectively aimed at Iran's nuclear-development industrial machinery, now considered a model for weapons of mass industrial destruction. (Note that Stuxnet was developed by a system-adversarial community, regardless of who they may be.) Moreover, leadership innovation and evolution is more prevalent in the adversary communities, motivated by opportunity, driven by reward, and enabled by open and pervasive information exchange. In contrast, the security community avoids information sharing, is insular in domain silos, and too focused on variations and refinements of existing tactical approaches. This is true in the bulk of systems-security research as well as in application.

What we have are adversarial communities functioning as natural systems, experimenting, innovating, and evolving in relentless progress to higher ecological superiority. In contrast we have systems-security communities and strategies that engineer deterministic systems, seek accuracy in risk analysis and security metrics, pursue trust and human compliance to policies and procedures at odds with personal and organization objectives, and expect standards to offer some sort of predictable protection. Adversarial communities operate as dynamic natural systems, preying upon static artificial systems. System resilience appears to be the only new promising concept with real attention, but as currently pursued that is only an acceptance of the tolerant prey role.

So now we come to the mechanisms at play, with an interest in the architecture of these mechanisms, the new vulnerabilities they bring with early stage engineered examples, and the ways in which architecture expresses security as an emergent property.

### **A Taste of New Thinking**

Both systems of systems and self-organizing security are still research topics in themselves. System-of-system (SoS) security is thus an even more elusive concept. Nevertheless, systems are becoming increasingly vulnerable, not because of any inherent security design flaw, but due to the context in which they operate. Both expected and unexpected interactions of

systems within SoS communities may have dramatic effects on security posture. In this issue, we examine several aspects of systems of systems and self-organizing security in an attempt to foster recognition of the security issues presented by the SoS in the systems engineering profession. Ten essays address this theme from various perspectives. Some serve as a wakeup call, some focus on the vulnerabilities of early systems of systems, and some begin to explore the lessons we can learn from self-organized security in systems of systems.

### **Systems of Systems and Security: A Defense Perspective**

Nowhere are the SoS security implications more clear than in the United States Department of Defense (DoD). Kristen Baldwin, principal deputy to the deputy assistant secretary of defense for systems engineering, together with Dr. Judith Dahmann, technical director, and Jonathan Goodnight, systems security engineer, open this issue with a revealing perspective on the current DoD approach to security and the challenges posed by systems of systems. Times have changed. Traditionally, “security has focused on keeping critical technology and information from ‘getting out.’ However, as DoD systems have come to depend on commercial technology and components that are increasingly sourced through complex global supply chains, a new security emphasis is emerging: keep malicious or negligent system elements or components from ‘getting in.’ Systems engineering and systems security engineering are applied to individual systems as part of the acquisition process. In most cases, systems engineering is not applied at the SoS level, and when it is, it has not emphasized systems security engineering for the SoS.” They then address the implications of the lack of attention to systems security engineering for systems of systems, and conclude that “addressing these implications provides both challenges and opportunities for the systems engineering community.”

### **Securing a System of Systems: Start with the Threats that Put the Mission at Risk**

Steve Sutton, recently retired as Technical Director at TASC, tackles the issue of system requirements, and suggests that threats to a system need to be a major consideration. Steve’s perspective is influenced by his work as cochair of the Anti-Terrorism International Working Group. He proposes some notional frameworks for guiding requirements development, and cites the action of the passengers on the fourth 9/11 flight as one example of self-organized security. He suggests how a national transportation system might incorporate self-organized security concepts. His essay suggests that “if we as systems engineers incorporate those recommendations and design the entire system from the beginning, we produce a system that performs the mission under all anticipated conditions.”

### **A Model-Based Approach to Support SoS Security Engineering for Data Policies**

Daniele Gianni and her four coauthors from the European Space Agency address the timely concept of a model-based approaches, and show examples of high-level security requirements derived from a data policy that become part of the agency’s SoS engineering activities, with diagrams. They go on to note, “systems managers can define the data policies depending on the organisational and political constraints. Security engineers can systematically derive the high-level security requirements from the model-based data policy specification. With the help of security engineers, SoS engineers can design the functional architecture by identifying the functional and non functional characteristics that meet the SoS requirements.” Included is a figure for security-requirements design and verification that shows how security requirements are tied to architecture.

## **SoS Issues in Security Engineering**

Jennifer Bayuk, program director of the Systems Security Engineering program at Stevens Institute of Technology School of Systems and Enterprises, cochair of the Systems Security Engineering working group, and author of several textbooks on systems security and control issues, takes the system and security engineering communities to task for the inadequacy of current standards and for their reliance on trust for the security of systems of systems. “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, and contractually obligate compliance with industry security standards rather than incorporate security as a design requirement for the emergent SoS.”

## **Trading Security and Safety Risks within SoS**

Dr. C. Warren Axelrod, senior consultant at Delta Risk, and author of two recent books on information security, takes on the complication that occurs when safety-critical and security-critical systems are combined in an SoS. Warren suggests that these SoS combinations are on the increase, and that “safety-critical systems introduce vulnerabilities for which security-critical systems are ill-prepared, and vice versa.” A table of differences contrasts security-critical with safety-critical systems, and then a revealing analytical example shows risk evaluations for these two types of systems when separate and when combined. “When safety-critical and security-critical software systems are connected to form SoS, not only might the various requirements assigned to either type of system not be consistent, they could actually conflict.” Warren closes with the suggestion that significant cross-training between engineers creating safety-critical systems and security-critical systems could result in a better outcome for both.

## **How Do We Measure “Security”?**

Dr. Fred Cohen, president of California Sciences Institute and CEO of Fred Cohen and Associates, is best known as the person who coined the term *computer virus*. Here, Fred’s electrical-engineering education prompts discomfort with the inability to measure security. He says, “If a systems engineer fails to take into account the operating temperature of space systems, the pressure requirements of underwater systems, or the radiation environment of nuclear safety systems, they get systems that fail when put into use in those environments. The same is true of the security environment. If the operating conditions will change over a range, we better be able to measure and design for them.” He notes that physical security and information security are not on equal footings here: “There are some pretty substantial notions underlying physical security and they are widely accepted. Information security has largely ignored these things, even though they could be quite helpful.”

## **Making People the Center of Systems Security**

Dr. Jennifer McGovern Narkevicius, managing director of Jenius, puts the spotlight on the human-centered issues of systems security. Cochair of the Human Systems Integration Working Group, Jennifer notes that “People as an element of a system fail when the system design does not take into account the capabilities and limitations of the human as an element of the system’s security. Human capabilities can significantly enhance systems security and their

limitations can significantly hamper the performance of those systems. Failure to include this basic understanding of such a central part of the system is the crux of the failure that is blamed on users.” She goes on to note issues and examples of social pressures, memory, and vigilance that affect the human role in systems security. She challenges systems engineers with many questions to consider, including “How can the design of the system leverage the strengths of the users? How can the system design suppress the weaknesses of the users?”

### **Security in the Social Age: A Systems Engineering Challenge**

John Ackley, team lead of the Malicious Code Analysis Group at CERT, part of the Software Engineering Institute at Carnegie Mellon University in Pittsburgh, Pennsylvania (US), addresses the oft-forgotten fact that humans are active components of many systems simultaneously, and the social systems that they are a part of cannot be walled off effectively from their involvement with technical systems. Humans thus form a nexus that crosses system boundaries. “Each of these nexus (for there are many humans in each system) bypasses what we usually understand to be the system boundary. This fact forces us to reconsider security, which has conventionally been achieved (or attempted) by fortifying system boundaries. Systems engineers cannot stop people from being social. Yet we have to design and build systems that are resilient to the risks posed by social networking. We cannot undo the viral success of online social networks, nor can we ignore the de facto SoS that swirl around each human. The human nexus problem makes our difficult job even harder, but we must account for it to succeed in delivering secure systems.” John discusses the role of incentives, usability, and the need to design for human error, one of the most underestimated sources of system risk.

### **Identifying Agile Security Patterns in Adversarial Stigmergic Systems**

Jena Lugosky, a Project Engineer for L-3 Communications, and Rick Dove of Stevens Institute of Technology and chair of the Systems Security Engineering Working Group, present an investigation of stigmergy as an organizing pattern in systems of systems. Stigmergy is a method for indirect communication in natural multiagent systems that influences collective behavior in ways typically called intelligent. This essay describes the four types of stigmergic coordination, develops a stigmergic interaction pattern for the Systems Security Engineering Working Group’s pattern project, relating the pattern to the Iraqi insurgency. They observe that, “This understanding may enable systems engineers, security engineers and decision makers to better understand and communicate the enemy’s methodology, and in so understanding, develop agile security systems that effectively address adversarial methods.”

### **Architectural Patterns for Self-Organizing Systems of Systems**

Craig Nicholson, a senior systems engineer for L-3 Communications, and Rick Dove present a six-element framework for self-organizing systems, and then relate that framework to the working group’s self-organizing systems-security pattern project. Six necessary characteristics are suggested: whole–part relationship, conditional dependency, common purpose, autonomy, situational awareness, and adaptability. Two SoS examples describe employment of self-organization patterns: the website Ushahidi and the terrorist organization Al Qaeda. Ushahidi shows the crowdsourced incident-reporting pattern using open communications for situational awareness, while Al Qaeda provides the use of the clear-cause coordination pattern to support security without the vulnerability of direct communications.

These essays are intended to spur interest and appreciation among the systems engineering community for the important role of security in the systems engineer's areas of responsibility. Systems engineers must recognize that systems security cannot be effective if it is not integrated intimately with the system architecture and the concept of operations. Moreover, systems security must be adaptable, evolvable, and proactively innovative, at least at the speed of adversarial innovation and evolution.