The Systems-of-Systems Challenge in Security Engineering

Vanea Chiprianov, Laurent Gallon, Manuel Munier, Philippe Aniorte, Vincent Lalanne
LIUPPA, Univ Pau & Pays Adour, France
Email: name.surname@univ-pau.fr

***Abstract*—Systems of systems (SoS) are large-scale systems composed of complex systems with difficult to predict emergent properties. One of the most significant challenges in the engineering of such systems is how to model and analyse their Non-Functional Properties, such as security. In this paper we identify, describe, analyse and categorise some challenges to security engineering of SoS. This catalogue of challenges offers a roadmap of major directions for future research activities, and a set of requirements against which present and future solutions of security for SoS can be evaluated.**

I. INTRODUCTION

Strategic attacks on a nation's infrastructure represent a great risk of disruption and loss of life and property. As the National Security Advisor, Condoleezza Rice, noted on 22 March 2001: 'US businesses, which own and operate more than 90% of the nation's banks, electric power plants, transportation systems, telecommunications networks, and other critical systems, must be as prepared as the government for the possibility of a debilitating attack in cyberspace.' Compounding the vulnerability of such systems is their interdependencies, because the impacts of attacks on one system can cascade into other systems [16].

As critical infrastructures are getting more and more dependant on Information Communication Technologies (ICT), the protection of these systems necessitates providing solutions that consider the vulnerabilities and security issues found in computers and digital communication technologies. However, the ICT systems that support these critical infrastructures are ubiquitous environments of composed heterogeneous components, and diverse technologies. These systems exhibit a variety of security problems and expose critical infrastructures to cyber attacks. Theses security challenges spread computer networks, through different ICT areas such as: cellular networks, operating systems, software, etc.

II. ENGINEERING OF SYSTEM-OF-SYSTEMS

Critical infrastructures have been considered a type of a larger class of systems, called Systems-of-Systems (SoS). SoS are large-scale concurrent and distributed systems that are comprised of complex systems [13]. Several definitions of SoS have been advanced, some of them are historically reviewed in [11] for example. SoS are complex systems themselves, and thus are distributed and characterized by interdependence, independence, cooperation, competition, and adaptation [7].

Examples of SoS comprise critical infrastructures like: electric grid interconnected with other sectors [23], the urban transportation sector interconnected with the wireless network [2], but also home devices integrated into a larger home monitoring system, interoperability of clouds [27], maritime security [22], embedded time-triggered safety-critical SoS [24], federated health information systems [6], communities of banks [3], self-organizing crowd-sourced incident reporting [20]. For example, a systematic review of SoS architecture [15] identifies examples of SoS in different categories of application domains: 58 SoS in defence and national security, 20 in Earth observation systems, 8 in Space systems, 6 in Modelling and simulation, 5 in Sensor Networking, 4 in Healthcare and electric power grid, 3 in Business information system, 3 in Transportation systems.

Characteristics that have been proposed to distinguish between complex but monolithic systems and SoS are [17]:

- *Operational Independence of the Elements*: If the SoS is disassembled into its component systems the component systems must be able to usefully operate independently. The SoS is composed of systems which are independent and useful in their own right.
- *Managerial Independence of the Elements*: The component systems not only *can* operate independently, they *do* operate independently. The component systems are separately acquired and integrated but maintain a continuing operational existence independent of the SoS.
- *Evolutionary Development*: The SoS does not appear fully formed. Its development and existence is evolutionary with functions and purposes added, removed, and modified with experience.
- *Emergent Behaviour*: The SoS performs functions and carries out purposes that do not reside in any component system. These behaviours are emergent properties of the entire SoS and cannot be localized to any component system. The principal purposes of the SoS are fulfilled by these behaviours.
- *Geographic Distribution*: The geographic extent of the component systems is large. Large is a nebulous and relative concept as communication capabilities increase, but at a minimum it means that the components can readily exchange only information and not substantial quantities of mass or energy.

Taking into account these characteristics specific to SoS needs specific engineering approaches. Most researchers agree that the SoS engineering approaches need to be different from the traditional systems engineering methodologies to account for the lack of holistic system analysis, design, verification,

validation, test, and evaluation [13], [5]. There is consensus among researchers [4], [18] and practitioners [1] that these characteristics necessitate treating a SoS as something different from a large, complex system. Therefore, SoS is treated as a distinct field by many researchers and practitioners.

III. Challenges in Security Engineering of Systems-of-Systems

Security engineering within SoS and SoS security life-cycle are influenced by SoS engineering and the SoS life-cycle. They need to take into account the characteristics specific to SoS, and how they impact security of SoS. At a general, abstract level, these impacts include [25]:

- *Operational Independence*: In an SoS, the component systems may be operated separately, under different policies, using different implementations and, in some cases, for multiple simultaneous purposes (i.e. including functions outside of the SoS purpose under consideration). This can lead to potential incompatibilities and conflict between the security of each system, including different security requirements, protocols, procedures, technologies and culture. Additionally, some systems may be more vulnerable to attack than others, and compromise of such systems may lead to compromise of the entire SoS. Operational independence adds a level of complexity to SoS that is not present in single systems.
- *Managerial Independence*: Component systems may be managed by completely different organisations, each with their own agendas. In the cyber security context, activities of one system may produce difficulties for the security of another system. What rights should one system have to specify the security of another system for SoS activities and independent activities? How can systems protect themselves within the SoS from other component systems and from SoS emerging activities? Does greater fulfilment require a component system to allow other component systems to access it?
- *Evolutionary Development*: An SoS typically evolves over time, and this can introduce security problems that the SoS or its components do not address, or are not aware of. Therefore, the security mitigations in place for an evolving SoS will be difficult to completely specify at design time, and will need to evolve as the SoS evolves.
- *Emergent Behaviour*: SoS are typically characterised by emerging or non-localised behaviours and functions that occur after the SoS has been deployed. These could clearly introduce security issues for the SoS or for its component systems, and therefore the security of the SoS will again need to evolve as the SoS evolves. In addition, responsibility for such behaviours could be complex and shared, leading to difficulties in deciding who should respond and where responses are needed.
- *Geographic Distribution*: An SoS is often geographically dispersed, which may cause difficulties in trying to secure the SoS as a whole if national regulations differ. These may restrict what can be done at different locations, and how the component systems may work together to respond to a changing security situation.

Identifying challenges to security engineering within SoS is the first step in engineering security within SoS. As highlighted by [18], a desirable research direction would be an integrated description and analysis method that can express and guarantee user level security, reliability, and timeliness properties of systems built by integrating large application layer parts - SoS. Moreover, systems engineering of defence systems and critical infrastructure must incorporate consideration of threats and vulnerabilities to malicious subversion into the engineering requirements, architecture, and design processes; the importance and the challenges of applying System Security Engineering beyond individual systems to SoS has been recognized [8]. Additionally, secure cyberspace has been recognized as one of the major challenges for 21st century engineering [26], [14].

Starting from the challenges related to characteristics specific to SoS, we further identify, describe and analyse challenges to security engineering of SoS. We organise them according to the activity of the security process in which they have the most impact. Of course, most challenges impact several activities, but for clarity purposes, we present them in the activity in which we consider they have the most impact.

A. Challenges impacting all Activities

Long life of SoS How to approach constraints associated with legacy systems? Consequently, will most SoS be composed of systems with uneven levels of 'system protection'?

B. Requirements Challenges

Identifying SoS security requirements How to identify these SoS overarching security requirements?

Security requirements modelling How can security be integrated into requirements modelling? How can a balance between near-term and long-term security requirements be achieved?

Ownership Who should have the ultimate ownership responsibility for the SoS? Who will be responsible for dealing with issues arising from the SoS, for example if the system was used for malicious purposes, who would be legally culpable? Who will be responsible for testing and proving the system is running as expected and fulfilling its security requirements?

Risk management How to identify and mitigate risks associated with end-to-end flow of information and control, without, if possible, focusing on risks internal to individual systems?

Holistic security Information security comprises: 1) Physical software systems security based on applying computer cryptography and safety or software criticality implementation; 2) Human / personnel security based on the procedure, regulations, methodologies that make an organisation / enterprise / system safe; 3) Cyber / Networking level that is mainly concerned with controlling cyber attacks and vulnerabilities and reducing their effects [19]. How can such holistic standards be extended to encompass SoS? How can they be applied and enforced in the context of SoS?

Requirements as source of variability How to adequately identify and allocate requirements to constituent systems for their respective teams to manage?

Security metrics for SoS What could be security-specific metrics and measures for an SoS? Is it possible to define a set of metrics which can be evaluated on the entire SoS, or are some security assessments limited to subparts of the SoS? Is it possible to define probability-theoretic metrics that can be associated with prediction models? How the mix of deterministic and uncertain phenomena, that come into play when addressing the behaviour of a SoS faced with malicious attacks, can be represented?

C. Design Challenges

Bridging the gap between requirements and design How to breach the gap between frameworks and implementation? How to assure a level of system and information availability consistent with stated requirements?

Designing security How can security be integrated into the SoS architecture? How to represent an exchange policy specification so as to verify some properties like: completeness, consistency, applicability and minimality?

Interdependency analysis How to identify threats that may appear insignificant when examining only first-order dependencies between composing systems of a SoS, but may have potentially significant impact if one adopts a more macroscopic view and assesses multi-order dependencies? How to asses the hidden interdependencies? How to represent the interdependencies existing among a group of collaborating systems? How such an approach can be integrated in a risk assessment methodology in order to obtain a SoS risk assessment framework? How to understand dependencies of a constituent system, on systems that are external to the formal definition of the SoS, but that nonetheless have security-relevant impacts to SoS capabilities?

New architectural processes Which would be the best suited process for architecting SoS and its security? Should it contain iterative elements, should it be agile, or model-based, etc? How does the type of dependencies between the development of SoS and the development of its constituent systems influence the design process of the SoS?

Design for evolution It is not sensible to assume that present security controls will provide adequate protection of a future SoS. Should there be a transition from system design principles based on establishing defensive measures aimed at keeping threats at bay, to postures that maintain operations regardless of the state of the SoS, including compromised states?

Scalability of security A larger number of users can interact with the SoS than with any of its composing systems. This means a possible increased number and/or scale of attacks. How can the security mechanisms for SoS be scaled up consequently?

Multiplicity of security mechanisms There are different security mechanisms at different levels. Defensive capabilities include for example physical security measures, personnel security measures, configuration control, intrusion detection, virus and mal-ware control, monitoring, auditing, disaster recovery, continuity of operations planning [10], cryptography, secure communications protocols, and key management methods that are time tested, reviewed by experts, and computationally sound [9]. How to use together effectively and efficiently all these mechanisms?

D. Implementation Challenges

Authentication The confirmation of a stated identity is an essential security mechanism in standalone systems, as well as in SoS. To achieve system interoperability, authentication mechanisms have to be agreed upon among systems to facilitate accessing resources from each system. How and when can this agreement be reached?

Authorisation In a SoS, users with different backgrounds and requirements should be granted accesses to different resources of each composing system. Therefore, a proper authorization mechanism is necessary for the composing systems to cooperate together and provide the best user experience possible for the SoS users [27]. How would delegation of rights be handled? Who would be responsible for it?

Accounting / Auditing In conjunction to security, accounting is necessary for the record of events and operations, and the saving of log information about them, for SoS and fault analysis, for responsibility delegation and transfer, and even digital forensics. Where will this information be tracked and stored and who will be responsible for the generation and maintenance of logs?

Non-Repudiation How can an evidence of the origin of any change to certain pieces of data be obtained in the context of an SoS? Who should collect these data, who can be trusted?

Encryption Encryption mechanisms should be agreed upon in order for SoS users from different endpoints to access the resources of a SoS. Cryptographic keys must be securely exchanged, then held and protected on either end of a communications link. This is challenging for a utility with numerous composing systems [9].

Security classification of data How to provide the ability to securely and dynamically share information across security domains while simultaneously guaranteeing the security and privacy required to that information? How to define multiple security policy domains and ensure separation between them?

Meta-data What kind of data should meta-data contain? What kind of meta-data should be legally-conformant to collect and employ? What kind of meta-data would technically be available? Should meta-data tags include data classification to provide controlled access, ensure security, and protect privacy? Should meta-data be crypto-bound to the original data to ensure source and authenticity of contents?

Heterogeneity and multiplicity of platforms How to detect cross-protocol, cross-implementation and cross-infrastructure vulnerabilities? How to correlate information across systems to identify such vulnerabilities and attacks?

E. Verification Challenges

Verifying the implementation satisfies the requirements When multiple, interacting components and services are involved, verifying that the SoS satisfies chosen security controls increases in complexity over standalone systems. This complexity is because the controls must be examined in terms of their different applications to the overall SoS, the independent composing systems, and their information exchange [12].

F. Release/Response Challenges

Configuration Who will be responsible for investigating any configuration issues and performing changes?

Monitoring Who will be responsible for monitoring addressing any faults or issues that may occur?

Runtime re-engineering In some cases, the SoS is only created at runtime, and the exact composition may not be known in advance. However, security currently takes time to establish, and there are many interrelated security issues that could create delay or loss of critical information. For some applications, runtime delays will have a big impact. Balance is therefore required in order to ensure security doesn't have a negative impact on operational effectiveness [21].

G. Possible Agenda for Tackling the Challenges

Following a Software Engineering approach, a possible agenda to tackle these challenges could be inspired from an iterative, incremental, V-like software development life-cycle. As such, a first step would consist in extracting and formulating requirements from the challenges. As these requirements could be divergent or even conflictual, several partial solutions could be expected to emerge. Therefore, in a second step, one or more architectural frameworks proposing an architecture for one or several software tools and processes to use them could be proposed. To validate and verify the requirements and the architecture(s), several test cases could be proposed. In a third step, the proposed framework(s) would be implemented in one or several programming languages. The fourth step would use the test cases to verify and validate the implementation. These steps would be repeted in an incremental way, until teh requirements are considered addressed.

IV. Conclusions

In this paper we provided a catalogue of challenges that have been identified in the literature regarding the subject of security engineering for Systems-of-Systems (SoS). Organised according to the security process activities, they represent an easy to consult, clear roadmap of major directions for future research. Future research can position their research questions according to the challenges identified here. Moreover, these challenges can serve as a set of requirements against which existing and future solutions to security engineering of SoS can be evaluated.

References

[1] Systems engineering guide for systems of systems, version 1.0., 2008.

[2] C. Barrett, R. Beckman, K. Channakeshava, Fei Huang, V.S.A. Kumar, A. Marathe, M.V. Marathe, and Guanhong Pei. Cascading failures in multiple infrastructures: From transportation to communication network. In *Critical Infrastructure, 5th Intl Conf on*, pages 1–8, 2010.

[3] W. Beyeler, R. Glass, and G. Lodi. Modeling and risk analysis of information sharing in the financial infrastructure. In R. Baldoni and G. Chockler, editors, *Collaborative Financial Infrastructure Protection*, pages 41–52. Springer, 2012.

[4] J. Boardman and B. Sauser. System of systems - the meaning of OF. In *System of Systems Eng, 2006 IEEE/SMC Intl Conf*, page 6 pp, 2006.

[5] R. T. Brooks and A. P. Sage. System of systems integration and test. *Information, Knowledge, Systems Management*, 5:261–280, 2006.

[6] M. Ciampi, G. Pietro, C. Esposito, M. Sicuranza, P. Mori, A. Gebrehiwot, and P. Donzelli. On Securing Communications among Federated Health Information Systems. In F. Ortmeier and P. Daniel, editors, *Computer Safety, Reliability, and Security*, volume 7613 of *LNCS*, pages 235–246. Springer, 2012.

[7] Cihan H. Dagli and Nil Kilicay-Ergin. *System of Systems Architecting*, pages 77–100. John Wiley & Sons, 2008.

[8] J. Dahmann, G. Rebovich, M. McEvilley, and G. Turner. Security engineering in a system of systems environment. In *Systems Conference, IEEE Intl*, pages 364–369, 2013.

[9] M. Duren, H. Aldridge, R. K. Abercrombie, and F. T. Sheldon. Designing and Operating Through Compromise: Architectural Analysis of CKMS for the Advanced Metering Infrastructure. In *The 8th Annual Cyber Security and Information Intelligence Research Wksh*, number 48, pages 1–3, 2013.

[10] D.L. Farroha and B.S. Farroha. Agile development for system of systems: Cyber security integration into information repositories architecture. In *IEEE Systems Conference*, pages 182 –188, 2011.

[11] A. Gorod, R. Gove, B. Sauser, and J. Boardman. System of systems management: A network management approach. In *System of Systems Engineering, IEEE Intl Conf. on*, pages 1–5, 2007.

[12] J. Hosey and R. Gamble. Extracting security control requirements. In *6th Wksh on Cyber Security and Info Intelligence Research*, 2010.

[13] M. Jamshidi. System of Systems - Innovations for 21st Century. In *Industrial and Information Systems, 3rd Intl Conf on*, pages 6–7, 2008.

[14] Roy S. Kalawsky. The Next Generation of Grand Challenges for Systems Engineering Research. *Procedia C. S.*, 16:834 – 843, 2013.

[15] J. Klein and H. van Vliet. A Systematic Review of System-of-systems Architecture Research. In *The 9th Intl ACM Sigsoft Conf on Quality of Software Architectures*, pages 13–22, 2013.

[16] S.J. Lukasik. Vulnerabilities and failures of complex systems. *Int. J. Eng. Educ.*, 19(1):206–212, 2003.

[17] M. W. Maier. Architecting principles for systems-of-systems. *Systems Engineering*, 1(4):267–284, 1998.

[18] M.W. Maier. Research Challenges for Systems-of-Systems. In *Systems, Man and Cybernetics, Intl Conf*, volume 4, pages 3149–3154, 2005.

[19] E.I. Neaga and M.J. de C Henshaw. Modeling the linkage between systems interoperability and security engineering. In *5th Intl Conf on System of Systems Engineering*, SoSE, 2010.

[20] C. Nichols and R. Dove. Architectural Patterns for Self-Organizing Systems-of-Systems. *Insight*, 4:42–45, 2011.

[21] C. E. Phillips, Jr., T.C. Ting, and S. A. Demurjian. Information sharing and security in dynamic coalitions. In *7th ACM Symposium on Access Control Models and Technologies*, pages 87–96, 2002.

[22] N. Ricci, A. M. Ross, and D. H. Rhodes. A Generalized Options-based Approach to Mitigate Perturbations in a Maritime Security System-of-Systems . *Procedia C. S.*, 16:718 – 727, 2013.

[23] S.M. Rinaldi, J.P. Peerenboom, and T.K. Kelly. Identifying, understanding, and analyzing critical infrastructure interdependencies. *Control Systems, IEEE*, 21(6):11–25, Dec 2001.

[24] F. Skopik, A. Treytl, A. Geven, B. Hirschler, T. Bleier, A. Eckel, C. El-Salloum, and A. Wasicek. Towards Secure Time-triggered Systems. In *Int Cf on Comp Safety, Reliability, and Security*, pages 365–372, 2012.

[25] A. Waller and R. Craddock. Managing runtime re-engineering of a System-of-Systems for cyber security. In *System of Systems Engineering (SoSE), 6th Intl Conf on*, pages 13–18, 2011.

[26] W. A. Wulf. Great achievements and grand challenges. Technical report, National Academy of Engineering, 2000.

[27] Z. Zhang, C. Wu, and D. W.L. Cheung. A Survey on Cloud Interoperability: Taxonomies, Standards, and Practice. *SIGMETRICS Perform. Eval. Rev.*, 40:13–22, 2013.