Challenges in Security Engineering of Systems-of-Systems

Vanea Chiprianov[1], Laurent Gallon[1], Manuel Munier[1], Philippe Aniorte[1] and Vincent Lalanne[1]

LIUPPA, Univ Pau & Pays Adour, France
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 analyze their Non-Functional Properties (NFP), such as security. In this review paper we identify, describe, analyze and categorize challenges to security engineering for SoS. This catalog of challenges offers a road-map 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.

1 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, with the result that impacts of attacks on one system can cascade into other systems [35].

As critical infrastructures are getting more and more dependent 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.

1.1 Engineering of System-of-Systems

Critical infrastructures are considered a type of a larger class of systems, Systems-of-Systems (SoS). SoS are large-scale concurrent and distributed systems, comprised of complex systems [22]. Several definitions of SoS have been advanced, some of them are historically reviewed in [18] for example. SoS are complex systems themselves, and thus are distributed and characterized by interdependence, independence, cooperation, competition, and adaptation [10].

Examples of SoS comprise critical infrastructures like: electric grid interconnected with other sectors [45], the urban transportation sector interconnected with the wireless network [3], but also home devices integrated into a larger home monitoring system, interoperability of clouds [55], maritime security [44], embedded time-triggered safety-critical SoS [48], federated health information systems [9], communities of banks [4], self-organizing crowd-sourced incident

reporting [42]. For example, a systematic review of SoS architecture [29] identifies examples of SoS in different categories of application domains: 58 SoS in defense and national security, 20 in Earth observation systems, 8 in Space systems, 6 in Modeling and simulation, 5 in Sensor Networking, 4 in Health-care 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 [36]:

- *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. They 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 Behavior*: The SoS performs functions and carries out purposes that do not reside in any component system. These behaviors 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 behaviors.
- *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.

Other sets of characteristics of SoS, partially overlapping, have been identified, e.g. [5]:

- *Autonomy*: The reason a system exists is to be free to pursue its purpose; this applies to both the whole SoS and constituent systems.
- *Belonging*: The component systems can choose to belong to the SoS based on their needs and enhance the value of the system's purpose.
- *Connectivity*: There has to be the means provided for the systems to communicate with each other for the exchange of information.
- *Diversity*: The SoS should be diverse and exhibit a variety of functions as a system compared to the limited functionality of the constituent systems.
- *Emergence*: The formation of new behaviors due to development or evolutionary processes.

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 [22], [8]. There is consensus among researchers [5], [37] and practitioners [2] 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, with its own conferences (e.g. IEEE International Conference on System of Systems Engineering, first one in 2006) and journals (e.g. International Journal of System of Systems Engineering).

As part of the SoS engineering initiative, an SoS life-cycle is essential. However, as underlined also by the *Evolutionary Development* characteristic for example, the SoS life-cycle depends on the degree of dependence between the SoS and its constituting systems [33]. If there are strong dependencies between the development of the SoS and the development of the

participating systems, they require synchronization between the construction and deployment of the composing systems and that of the SoS. For example, such SoS are regional health information organizations and intelligent transportation systems. On the other hand, when missing elements are not critical, or when alternative solutions exist, SoS development can proceed without waiting. For example, data mash-ups - web applications that compose services - are such SoS. However, for these SoS is even more difficult to ensure their NFPs, exactly because their dependencies from their composing systems are looser.

1.2 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 [52]:

- *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 organizations, 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 fulfillment 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 Behavior*: SoS are typically characterized by emerging or non-localized behaviors 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 behaviors 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.

And for the other set of characteristics [18]:

- *Autonomy*: Ensuring security of component systems.
- *Belonging*: Restricting/allowing component systems access to SoS.
- *Connectivity*: Protecting from unauthorized integration.
- *Diversity*: Restricting/allowing diverse behavior.
- *Emergence*: Preventing/managing policies "bad" emergent behavior.

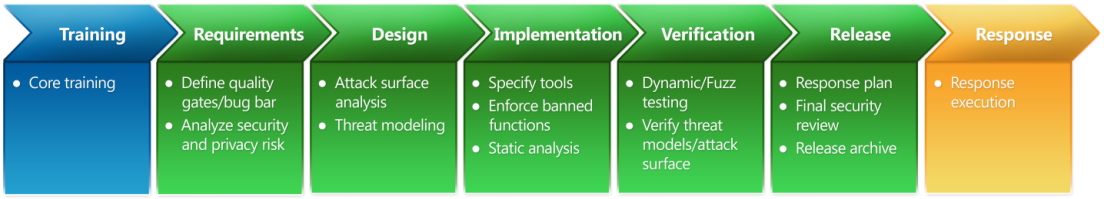

Figure 1: Microsoft Security Development Life-cycle, [20]

The specificities of security engineering for SoS, identified by analyzing the impact of SoS-specific characteristics, run of course much deeper. They touch all phases of the SoS life-cycle and all phases of the security engineering process. A generic security process is for example that proposed by Microsoft, Fig. 1. It comprises the phases of: training, requirements, design, implementation, verification, release, and response. A process for SoS security engineering has been proposed e.g. by [7]. This consists in the following activities, which can be largely concurrent and iterative: information gathering, flow analysis, security test and evaluation and end-to-end testing, target architecture and transition planning, security of SoS modeling, SoS security policy developing and risk management. In this paper, we further identify challenges to security engineering within SoS and organize them according to the security process.

Identifying challenges to security engineering within SoS is the first step in engineering security within SoS. As highlighted by [37], a very 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 defense 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 [11]. Additionally, secure cyberspace has been recognized as one of the major challenges for 21st century engineering [53], [24].

2 Challenges in Security Engineering of Systems-of-Systems

Starting from the challenges related to characteristics specific to SoS, we further identify, describe and analyze challenges to security engineering of SoS. We organize 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.

2.1 Challenges impacting all Activities of the Security Process

Long life of SoS Most composing systems of a SoS are used in a context with other systems that have not necessarily gone through the same security engineering processes. Even if all the systems in a given composition had gone through the same security engineering process individually, composing them to achieve a new capability has the potential for new vulnerabilities and threats, and thus new risks to the end-to-end SoS. And the number of new systems organizations (e.g. Department of Defense) fields is small compared to the number of legacy ones. How to approach constraints associated with these legacy systems [7]? Consequently, will

most SoS be composed of systems with uneven levels of system protection [11]?

2.2 Requirements Challenges

Identifying SoS security requirements Because requirements are taken on by the constituent systems to meet the SoS objectives, identifying the security requirements for the overarching SoS provides a framework for assessing the adequacy of the system security engineering actions on the part of the constituent systems for security for the SoS and its mission [11]. How to identify these overarching security requirements?

Security requirements modeling SoS security engineering involves a tension between near-term risk mitigation and long-term evolution to a more secure SoS architecture. In the near term, risks can usually be mitigated effectively by controls at policy domain boundaries and at interfaces between individual systems. In the long term, uniform enforcement mechanisms within and between policy domains not only mitigate risks more effectively but also improve interoperability and maintainability. How can security be integrated into requirements modeling [7]? 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 [27]?

Risk management This is concerned with management and control for the assessment, updating and mitigating of risks [27]. Security-related risks would be part of SoS risk identification and mitigation. They include new security risks resulting from new SoS capabilities composed from interacting constituent systems, as well as any residual security risk of constituent systems [11]. 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 [7]? While there are standards for risk management of standalone systems [21], there are not for SoS. To what extent do they apply to SoS; should such standards be extended to SoS?

Security of interoperability Several important aspects of enterprise interoperability have been the focus of European research programmes and initiatives such as European Interoperability Framework (EIF), INTEROP-Vlab and ATHENA-IP. How should these interoperability approaches consider organizational and human factors, such as personal responsibilities (policies and best practice for system security) from the earliest stage of the analysis? How should they address information protection, trust and security [41]?

Holistic security For many years the focus has been mainly on IT security (e.g. cryptographic analysis) and usually the implementation of security tools has been done by IT departments and computer experts. During the early 90s key aspects started to change and the first draft of an information security management standard BS 7799 [1] was produced. It focused on security related to synergies of people, processes, and information as well as IT systems. Since then, early security management standards have been transformed into international standards published by ISO/IEC [21]. These standards are being used by enterprises and organizations, and there are currently several initiatives holistically capturing security issues related to people, organization and technology. However application of standards does not guarantee solving

the broad spectrum of systems security problems. 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 organization / enterprise / system safe; 3) Cyber / Networking level that is mainly concerned with controlling cyber attacks and vulnerabilities and reducing their effects [41]. 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 It may be difficult to identify a specific SoS configuration that will meet the objectives and quantify the requirements allocated for each specific composing system. Requirements are often the largest source of variability and unknown quantities at the start of traditional system engineering projects. From a SoS perspective, increased difficulty is added with multiple systems and independently managed requirements teams whose efforts need to be coordinated in order to achieve the SoS objectives. How to adequately identify and allocate requirements to constituent systems for their respective teams to manage [16]?

Security metrics for SoS What could be security-specific metrics and measures for an SoS [11]? 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 behavior of a SoS faced with malicious attacks, can be represented [47]?

Balancing costs and gains of information sharing Information sharing participation carries with it costs which need to be balanced by direct expected gain or to be subsidized in order to have a critical number of composing systems to agree to share information and to discourage free riding. Agreeing to share information entails some cost to the participating system: these include costs of acquiring and maintaining equipment, training staff to use it, and integrating it into existing business practices. Although these expenses may be small in relation to existing operating costs, non-monetary costs might have some influence on the decision to participate. The benefits of information sharing include a decreased probability that a particular attack will be successful, and an increased rate of detection and recovery should an attack succeed. A large part of this benefit is naturally seen by the composing systems; however there may be important positive externalities as well. If an attack or even an (in)voluntary input error on a particular system is successful, it may create problems for other systems in the SoS. If the attack introduces propagating malware, for example, it might be spread to other systems through business or social communications. Operational disruptions in one system might impose costs on other systems by preventing clearing and settlement of inter-system transactions or client transactions. There are also possible reputation costs to the SoS as a whole arising from a successful attack on a single system. Such externalities create an incentive for each system in the SoS to see other systems join in an information sharing arrangement. Conversely some of the benefits created by a joining system are experienced by other systems in the SoS, whether or not they themselves participate [4].

2.3 Design Challenges

Bridging the gap between requirements and design The use of standards and architectural frameworks does not always guarantee achievement of desired levels of interoperability

security. How to breach the gap between frameworks and implementation [41]? How to assure a level of system and information availability consistent with stated requirements [15]?

Designing security Security is often taken into account only at the end of the development life-cycle of a system. Consequently, any a posteriori modification is very expensive. That is why, security must be taken into account as soon as possible and designed-in rather than relying on hardening of systems post implementation [30]. Moreover, SoS must be designed for robustness under planned malicious assault [35]. But how can security be integrated into the SoS architecture [7]? How to represent an exchange policy specification so as to verify some properties like: completeness, consistency, applicability and minimality [12]?

SoS security modeling and (continuous) analysis The protection of SoS requires a better understanding of how control needs to be designed in a top-down design approach in order to cover all the aspects of the composing systems. High interconnectivity, complexity, and dependency in SoS are spread through multiple levels and evolve over time with unpredictable behavior. Therefore, it is hard to understand, design, and manage composing systems [38]. How to represent the SoS in a form that lends itself to detailed analysis, especially when full details of the component systems may not be readily available? SoS analysis can drive changes in the composing systems to exploit opportunities or correct problems that were not originally anticipated. A key architectural tool in this respect may be the use of predictive modeling and simulation to compare architectural alternatives. In any process for improving a SoS, alternative architectures would need to be carefully considered and modeled to ensure the SoS is not compromised or undesirable emergent behaviors result. Some of the elements that comprise a SoS may be outside the control of other elements in the SoS. This means the SoS may not be responsive to a single analysis. Should, therefore, SoS analysis be incremental and the SoS should be available for testing almost on a continuous basis [25]?

Interdependency analysis It is concerned with examining the possible cumulative effects (escalation or cascading) of a single security incident on multiple systems. Such cascading failures could result in a complete blackout. Hence, it is important to identify cascading vulnerabilities before they happen [38]. 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 [31]? 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 [17]? 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 [11]?

Security agreements How to manage security protections and security risk acceptance relationships among multiple systems? How about also supporting SoS evolution in terms of authorization to connect a constituent system to some other constituent system; the handling of risks propagated from one constituent system to some other constituent system of the SoS? How about the distribution and inheritance of security measures; assessment processes; authorization and acceptance processes; the roles and responsibilities for operation and sustainment of protections necessitated by SoS capabilities [11]?

New architectural processes Governance plays a significant role; designing SoS needs to be addressed different from the traditional process of stove-piped systems [15]. Which would be the best suited process for architecting SoS? 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?

Defensive design SoS must be designed defensively, and this holds for composing systems as well. Defensive design is nothing new in areas involving physical processes. Aircraft engines and canopies are designed to withstand bird impacts, and structures are designed for the one-hundred year storm and magnitude 8 earthquakes. In ICT there seems to be no equivalent stressing design criterion, and such systems are easily overwhelmed by nothing more sophisticated than just driving up the rates at which system resources are accessed [35].

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 complex defensive measures aimed at keeping threats at bay, to postures that maintain operations regardless of the state of the SoS, including compromised states [14]? Should the focus move from avoiding threats from adversary action, over which we have limited control, to the larger, more inclusive goal of controlling SoS vulnerabilities [54]?

Security assurance cases How to model security assurance cases of SoS? While a system may be deemed secure by itself in a particular configuration, the introduction of other systems into that environment increases the complexity of the assurance model and must be considered and evaluated as part of a larger system [30].

Geographical aspects How should data be conveyed by geographical proximity? Like in wide area networks? Data that are related could be stored in different repositories belonging to several distinct systems, each with different security assurances [9].

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 [9]. Therefore, the security mechanisms for the security of SoS should 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 [15], cryptography, secure communications protocols, and key management methods that are time tested, reviewed by experts, and computationally sound [14]. How to use together effectively and efficiently all these mechanisms [9]?

2.4 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? Two kinds of authentication mechanisms are commonly presented: HTTP Authentication and Public Key Infrastructure [55].

Without authentication mechanisms to limit access, there is limited protection for the integrity of the information being transmitted [38].

What mechanism would allow various identity systems to inter-operate, so as all identity providers, relying parties (identity consumers) and subjects (users) work together using existing systems. Component systems may be developed by anyone; no single party has control. This mechanism should ensure: 1) consent: location systems must only reveal a user's location information with the user's consent; 2) minimal disclosure: location systems should reveal only the location information necessary; 3) granularity: location providers should specify all the levels of granularity of location information they are able to provide, and location consumers should specify all the levels of location information they are able to consume and switch between providers when one is shut down or temporarily unavailable [13].

Time constraints One of the main prerequisites for security of real-time SoS is that devices properly mutually authenticate themselves to prevent insertion of malicious devices or messages in case of a man-in-the middle attack. The main challenge in the design and implementation of device authentication mechanisms is to retain the temporal properties of a real-time system, i.e., the designer has to take care that introducing an authentication scheme in the real-time communication does not spoil the original real-time properties of the time-triggered system. Any additional and unpredictable delay in the communication path is critical for the communication and consequently for the access control and traffic separation based on the time-triggered protocol [48]. Of course, the authentication case can be generalized to other security mechanisms that may introduce delays in time-constrained SoS.

Authorization It is concerned with the management and control of the authorization schemes used and the ability to grant SoS authentication to interested parties [27]. Systems need to allocate the resources or rights according to a user's credentials after the user has proven to be what they stated. 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 [55]. 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. Interoperability among systems can be seriously affected if no such information is available. However, there is no well-defined best-practice guideline (not necessarily standard) on accounting agreed upon and adopted [55]. Where will this information be tracked and stored and who will be responsible for the generation and maintenance of logs [27]? How could this be reconciled with privacy concerns for example?

Non-Repudiation It is particularly important in a SoS where systems are legally bound by certain contracts. Verifying that one of the systems has indeed performed a certain action becomes necessary. How can an evidence of the origin of any change to certain pieces of data be obtained in the context of an SoS [9]? Who should collect these data, who can be trusted?

Encryption Challenges arise when ensuring the security between SoS endpoints through communication encryption. Without encryption to protect data as it flows through these insecure

connections, there is limited protection for the integrity of the information being transmitted [38]. Encryption mechanisms should be agreed upon in order for SoS users from different end-points to access the resources of a SoS. Encryption mechanisms like SSL, TSL, VPN are some of the widely adopted protocols [55]. 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 [14].

Cryptographic key management Are the current trust models for cryptographic systems appropriate for SoS? The hierarchical trust model on which Public Key Infrastructure systems depend on is only as strong as the keys and trust points near the top of the pyramid (i.e. the keys used to issue certificates). As the SoS increases in size, the potential impact of a root compromise event also increases, particularly as the SoS crosses organizational boundaries. In the case of a Certificate Authority compromise, all systems that have the compromised certificate in their certificate stores are susceptible to compromise [14].

On the other hand, various trust management systems and associated trust models are being introduced, customized according to their target applications. The heterogeneity of trust models may prevent exploiting the trust knowledge acquired in one context in another context although this would be beneficial for the overall SoS. How to achieve interoperability between heterogeneous trust management systems [46]?

Security classification of data SoS may have multiple security domains [26]. In each composing system, data may have a specific security classification. It is possible that when combining two different sources at different classification levels, the new synthesized SoS product results with yet an additional classification - this necessitates a cross-domain solution [49]. 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 [15]? How to define multiple security policy domains and ensure separation between them? At the boundary between domains on different systems, information is often handed off from one set of enforcement mechanisms to another. Inconsistencies between policies and enforcement mechanisms frequently create vulnerabilities at policy boundaries which must be addressed by SoS security engineering [7].

Composing control policies How to express security and information assurance control into a uniformly, verifiable form so that they can be easily composed to form functional security requirements [19] [51]? In a top-down approach, how to flow down security policies to lower levels in the program? This enables specific interpretation of policy to different levels: work-packages, operational focus areas and individual projects, i.e. the top-level SoS goals need to be instantiated in and tailored to the high-level objectives for each system [30].

Context-based policies Composing systems might not be comfortable disclosing sensitive data to other entities except under certain conditions including transient conditions at the time of access. This shared data should be accessed exclusively by authorized parties, which may vary depending on the context (e.g. in emergency situations, or based on the location of the requester) [50]. In many cases it is the context-based policy that drives the data sharing while the number or recipients or their identities may not be known in advance. Interestingly, it is not just the data that is sensitive but also the policies for sharing the data. Therefore, there may be a need for policy-based data encryption techniques that support: 1) multiple recipients, 2) data and policy secrecy and 3) context-based policy enforcement [6].

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 [15]?

Heterogeneity and multiplicity of platforms How to detect cross-protocol, cross-implementation and cross-infrastructure vulnerabilities? These vulnerabilities may be created for example when bridging two types of networks, e.g. VoIP and PSTN. How to correlate information across systems to identify such vulnerabilities and attacks [28]?

Hardware-enabled security Devices and systems that can place trust in a hardware mechanism to ensure operational integrity, force attacks to physically compromise a device in order to successfully perpetrate an attack. This provides a significant mechanism that devices can use to not only detect compromise, but also manage it and recover from it [14]. This would certainly benefit the security of the SoS as well.

2.5 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 [19].

2.6 Release/Response Challenges

Configuration It is related to managing and altering security configuration settings associated with the SoS. Who will be responsible for investigating any configuration issues and performing changes [27]?

Monitoring It deals with monitoring for faults and issues within the SoS and ultimately who will be responsible for addressing any issues that may occur [27]?

Operational environment It deals with the control and assessment of the operational environment and the creation and enforcing of policies to control environmental security related to the SoS [27].

Runtime re-engineering In some cases, the SoS is only created at runtime, and the exact composition may not be known in advance. In such cases, it is difficult to fully plan and design the security of an SoS as part of the pre-deployment design. 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 [43]. Moreover, in some cases, evolution of SoS may impact the conformance to requirements. At runtime, it is important to ensure that the current required level of security is achieved. If not, then re-engineering is required to resolve the situation. As part of the re-engineering, it is important to monitor and assess the SoS security state in order to determine

the nature of any security inadequacies, choose an appropriate course of action to resolve the deficiencies and implement it [52].

3 Related Work

A survey [27] examines selected approaches for the provision of security within SoS. The survey identifies some challenges to SoS security, like ownership, auditing, configuration, monitoring, authorization, risk management, operational environment, but focuses mainly on the running and operation of the SoS, which corresponds mainly to the Release/Response activities in the security process. We take into account all activities of the security process and identify a greater number of challenges related to each of them.

A systematic review of SoS architecture research [29] identifies 14 studies that discuss security of SoS. However, it does not identify the challenges to SoS security, and it does not analyze in further detail these studies.

The paper [11] identifies a number of challenges, issues to security of defense SoS. However, it is focused on defense SoS, while we take into account all types of SoS.

The paper [26] presents a framework for secure SoS composition, with a substantial related work. However, it focuses on solutions, not on challenges.

4 Conclusions and Future Work

In this review paper we have provided a catalog of challenges that have been identified in the literature regarding the subject of security engineering for Systems-of-Systems (SoS). Organized according to the security process activities, they represent an easy to consult, clear road-map 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.

Concerning our own work in the field of security engineering for SoS, we are tackling the **Security requirements modeling** and the **SoS security modeling and analysis** challenges with Model Driven Engineering (MDE) approaches by proposing Domain Specific Modeling Languages (DSML) both at the Computer Independent Model and at the Platform Independent Model levels, based on our works for systems [39]. Model Driven Security has more than a decade of existance, with major approaches reviewed e.g. in [34]. Using MDE also offers the advantage of naturally tackling the **Bridging the gap between requirements and design** challenge with model transformations between the DSML for requirements and the one for design.

The security aspects modeled in the DSMLs are in the case of our work mainly related to **Authorization**. In this direction, we are also investigating how to express and compose context-based access control policies: **Composing control policies** and **Context-based policies**.

A complementary research direction we are pursuing deals with **Risk management**, looking at how new threats and vulnerabilities, and thus new risks can be identified and further refined and accounted for - **Accounting / Auditing**. This is based on our previous work about information security risk management for information systems interconnected through services [32]. Moreover, a survey of the aspects related to data risks in business processes composed through the cloud was presented in [23]. The data, and **Meta-data** needed and legally permitted for accounting is another aspect we investigate [40].

Acknowledgments

The authors of this paper particularly thank the authors of the systematic review [29] for providing the list of the studies they have found to deal with security of SoS.

References

[1] BS 7799:Part 1:1995 information security management code of practice for information security management systems. Technical report, BSI British Standards, 1995.

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

[3] 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.

[4] Walter Beyeler, Robert Glass, and Giorgia Lodi. Modeling and risk analysis of information sharing in the financial infrastructure. In Roberto Baldoni and Gregory Chockler, editors, *Collaborative Financial Infrastructure Protection*, pages 41–52. Springer Berlin Heidelberg, 2012.

[5] J. Boardman and B. Sauser. System of systems - the meaning of OF. In *System of Systems Engineering, 2006 IEEE/SMC International Conference on*, pages 6 pp.–, April 2006.

[6] Rakesh Bobba, Himanshu Khurana, Musab AlTurki, and Farhana Ashraf. Pbes: a policy based encryption system with application to data sharing in the power grid. In *4th International Symposium on Information, Computer, and Communications Security*, ASIACCS, pages 262–275, 2009.

[7] D.J. Bodeau. System-of-systems security engineering. In *Computer Security Applications Conference, 1994. Proceedings., 10th Annual*, pages 228–235, Dec 1994.

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

[9] Mario Ciampi, Giuseppe Pietro, Christian Esposito, Mario Sicuranza, Paolo Mori, Abraham Gebrehiwot, and Paolo Donzelli. On Securing Communications among Federated Health Information Systems. In Frank Ortmeier and Peter Daniel, editors, *Computer Safety, Reliability, and Security*, volume 7613 of *LNCS*, pages 235–246. Springer, 2012.

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

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

[12] Remi Delmas and Thomas Polacsek. Formal methods for exchange policy specification. In Camille Salinesi, MoiraC. Norrie, and scar Pastor, editors, *Advanced Information Systems Engineering*, volume 7908 of *Lecture Notes in Computer Science*, pages 288–303. Springer, 2013.

[13] Nick Doty. The case for a location metasystem. In *2nd International Workshop on Location and the Web*, LOCWEB, 2009.

[14] Michael Duren, Hal Aldridge, Robert K. Abercrombie, and Frederick T. Sheldon. Designing and operating through compromise: Architectural analysis of ckms for the advanced metering infrastructure. In *Proceedings of the Eighth Annual Cyber Security and Information Intelligence Research Workshop*, CSIIRW '13, pages 48:1–48:3, New York, NY, USA, 2013. ACM.

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

[16] David Flanigan and Peggy Brouse. Evaluating the allocation of border security system of systems requirements. *Procedia Computer Science*, 16(0):631 – 638, 2013. Conf on Systems Eng Research.

[17] I.N. Fovino and M. Masera. Emergent disservices in interdependent systems and system-of-systems. In *IEEE Intl Conf on Systems, Man and Cybernetics*, volume 1, pages 590–595, 2006.

[18] A. Gorod, R. Gove, B. Sauser, and J. Boardman. System of systems management: A network management approach. In *System of Systems Engineering, 2007. SoSE '07. IEEE International Conference on*, pages 1–5, April 2007.

[19] J. Hosey and R. Gamble. Extracting security control requirements. In *6th Annual Workshop on Cyber Security and Information Intelligence Research*, CSIIRW, 2010.

[20] Michael Howard and Steve Lipner. *The security development lifecycle*. O'Reilly Media, 2009.

[21] ISO/IEC. ISO/IEC 27005:2011: Information security risk management. Technical report, International Organization for Standardization (ISO), Geneva, Switzerland, 2011.

[22] M. Jamshidi. System of Systems - Innovations for 21st Century. In *Industrial and Information Systems, 2008. ICIIS 2008. IEEE Region 10 and the Third intl Conf on*, pages 6–7, Dec 2008.

[23] Elena Jaramillo, Manuel Munier, and Philippe Aniorte. Information security in business intelligence based on cloud: A survey of key issues and the premises of a proposal. In *WOSIS*, 2013.

[24] Roy S. Kalawsky. The Next Generation of Grand Challenges for Systems Engineering Research. *Procedia Computer Science*, 16(0):834 – 843, 2013. Conf. on Systems Engineering Research.

[25] Roy S. Kalawsky, D. Joannou, Y. Tian, and A. Fayoumi. Using architecture patterns to architect and analyze systems of systems. *Procedia Computer Science*, 16(0):283 – 292, 2013. 2013 Conference on Systems Engineering Research.

[26] M. Kennedy, D. Llewellyn-Jones, Q. Shi, and M. Merabti. A framework for providing a secure system of systems composition. In *The 12th Annual Conference on the Convergence of Telecommunications, Networking & Broadcasting (PGNet 2011)*, 2011.

[27] Michael Kennedy, David Llewellyn-Jones, Qi Shi, and Madjid Merabti. System-of-systems security: A survey. In *The 11th Annual Conference on the Convergence of Telecommunications, Networking & Broadcasting (PGNet 2010)*, 2010.

[28] A.D. Keromytis. A comprehensive survey of voice over ip security research. *Communications Surveys Tutorials, IEEE*, 14(2):514–537, Second 2012.

[29] John Klein and Hans van Vliet. A systematic review of system-of-systems architecture research. In *Proceedings of the 9th International ACM Sigsoft Conference on Quality of Software Architectures*, QoSA '13, pages 13–22, New York, NY, USA, 2013. ACM.

[30] R. Koelle and M. Hawley. Sesar security 2020: How to embed and assure security in system-of-systems engineering? In *Integrated Communications, Navigation and Surveillance Conference (ICNS), 2012*, pages E8–1–E8–11, April 2012.

[31] Panayiotis Kotzanikolaou, Marianthi Theoharidou, and Dimitris Gritzalis. Interdependencies between critical infrastructures: Analyzing the risk of cascading effects. In Sandro Bologna, Bernhard Hmmerli, Dimitris Gritzalis, and Stephen Wolthusen, editors, *Critical Information Infrastructure Security*, volume 6983 of *Lecture Notes in Computer Science*, pages 104–115. Springer, 2013.

[32] Vincent Lalanne, Manuel Munier, and Alban Gabillon. Information security risk management in a world of services. In *PASSAT*, 2013.

[33] G. Lewis, E. Morris, P. Place, S. Simanta, D. Smith, and L. Wrage. Engineering systems of systems. In *Systems Conference, 2008 2nd Annual IEEE*, pages 1–6, April 2008.

[34] Levi Lucio, Qin Zhang, Phu Hong Nguyen, Moussa Amrani, Jacques Klein, Hans Vangheluwe, and Yves Le Traon. Advances in model-driven security. *Advances in Computers*, 93:103–152, 2014.

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

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

[37] M.W. Maier. Research challenges for systems-of-systems. In *Systems, Man and Cybernetics, 2005 IEEE International Conference on*, volume 4, pages 3149–3154, Oct 2005.

[38] M. Merabti, M. Kennedy, and W. Hurst. Critical infrastructure protection: A 21st century challenge. In *Communications and Information Technology (ICCIT), 2011 International Conference*

on, pages 1–6, March 2011.

[39] D. Munante, L. Gallon, and P. Aniorte. An approach based on model-driven engineering to define security policies using orbac. In *Availability, Reliability and Security (ARES), 2013 Eighth International Conference on*, pages 324–332, Sept 2013.

[40] Manuel Munier, Vincent Lalanne, Pierre-Yves Ardoy, and Magali Ricarde. Legal issues about metadata: Data privacy vs information security. In *DPM*, 2013.

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

[42] Craig Nichols and Rick Dove. Architectural patterns for self-organizing systems-of-systems. *Insight*, 4:42–45, 2011.

[43] Charles E. Phillips, Jr., T.C. Ting, and Steven A. Demurjian. Information sharing and security in dynamic coalitions. In *Proceedings of the Seventh ACM Symposium on Access Control Models and Technologies*, SACMAT '02, pages 87–96, New York, NY, USA, 2002. ACM.

[44] Nicola Ricci, Adam M. Ross, and Donna H. Rhodes. A generalized options-based approach to mitigate perturbations in a maritime security system-of-systems. *Procedia Computer Science*, 16(0):718 – 727, 2013. 2013 Conference on Systems Engineering Research.

[45] 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.

[46] Rachid Saadi, MohammadAshiqur Rahaman, Valrie Issarny, and Alessandra Toninelli. Composing trust models towards interoperable trust management. In Ian Wakeman, Ehud Gudes, Christian-Damsgaard Jensen, and Jason Crampton, editors, *Trust Management V*, volume 358 of *IFIP Advances in Information and Communication Technology*, pages 51–66. 2011.

[47] Luca Simoncini. Dependable and historic computing. chapter Socio-technical Complex Systems of Systems: Can We Justifiably Trust Their Resilience?, pages 486–497. Berlin, Heidelberg, 2011.

[48] Florian Skopik, Albert Treytl, Arjan Geven, Bernd Hirschler, Thomas Bleier, Andreas Eckel, Christian El-Salloum, and Armin Wasicek. Towards secure time-triggered systems. In *Proc of the 2012 Intl Conf on Computer Safety, Reliability, and Security*, pages 365–372. Springer, 2012.

[49] M.A. Solano. Sose architecture principles for net-centric multi-int fusion systems. In *6th International Conference on System of Systems Engineering*, SoSE, pages 61 –66, June 2011.

[50] Daniel Trivellato, Nicola Zannone, and Sandro Etalle. Poster: protecting information in systems of systems. In *18th ACM Conf on Computer and Communications Security*, pages 865–868, 2011.

[51] Daniel Trivellato, Nicola Zannone, Maurice Glaundrup, Jacek Skowronek, and Sandro Etalle. A semantic security framework for systems of systems. *Int. J. Cooperative Inf. Syst.*, 22(1), 2013.

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

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

[54] William Young and Nancy G. Leveson. An integrated approach to safety and security based on systems theory. *Commun. ACM*, 57(2):31–35, February 2014.

[55] Zhizhong Zhang, Chuan Wu, and David W.L. Cheung. A survey on cloud interoperability: Taxonomies, standards, and practice. *SIGMETRICS Perform. Eval. Rev.*, 40(4):13–22, April 2013.