Development of Secure System of Systems Needing a Rapid Deployment

Nan Messe, Nicolas Belloir, Vanea Chiprianov, Imane Cherfa, Régis Fleurquin, Salah Sadou

Development of Secure System of Systems Needing a Rapid Deployment

Nan Messe*, Nicolas Belloir*†, Vanea Chiprianov*, Imane Cherfa*‡, Régis Fleurquin*, Salah Sadou*
* *Université Bretagne Sud - IRISA, France*
† *CREC St Cyr, France*
‡ *University of Blida - LRDSI, Algeria*
Email: firstname.lastname@irisa.fr

***Abstract*—In certain cases, such as secure humanitarian corridors in a conflict zone, a special type of SoS, needing a rapid deployment, has to be developed. Because of the tense time constraint, usually only a domain expert is responsible with this development. However, many such SoSs also have to take into account the security aspect. How to help a domain expert integrate the security aspect into the rapid development of an SoS? In this proposal paper, we present an approach and a tool suite that help the domain expert tag business assets using security properties, which are then used to identify vulnerabilities and to propose possible security control mechanisms. We illustrate our proposal on a case study.**

***Index Terms*—security, security model and architecture design, meta-modelling, domain model, causal chain**

I. Introduction

In this paper, we deal with a special type of System-of-systems (SoS): SoS needing a Rapid Deployment (SoSnRD) such as a secure rescue operation after an earthquake or a military operation. This kind of system has to be implemented as quickly as possible (within a few hours or days) to respond to an emergency. To build SoSnRD, we cannot rely on traditional SoS development methodologies, such as NAF [19], COMPASS [20] or DANSE [15]. These methodologies usually assume that: i) there is a significant amount of time to perform the development (months or years) and ii) a potentially large number of stakeholders can be involved. However, the design of an SoSnRD is often the responsibility of a single person: a domain expert. Such experts have significant experiences in setting up such SoSs in their fields of intervention. Thus, they are often able to imagine a preliminary design solution very quickly: essentially adapting a 'generic' solution to a particular context.

As an SoSnRD may be deployed in a hostile environment, it is essential to take into account the security aspect. Unfortunately, the domain experts generally do not have any security skills. They cannot rely on existing SoS security methodologies such as SoSSec [12]. Indeed, such methods require a long time and interactions with security experts, which is not always possible when in an emergency. Accordingly, these experts can propose a solution that meets functional requirements, but not one that integrates any security aspect.

In this paper, we propose a prospective vision on the development of SoSnRD integrating the security aspect. We propose an assistance that will enable a domain expert who is not a security specialist to document and integrate security in his/her design throughout all the architecture design process. The core of this assistance promotes the concept of 'asset'. We will show that assets can help bridge the gap between domain expert's knowledge and security concerns.

In the remainder of this paper, section II presents examples of SoSnRD and highlights the particular challenges associated with the secure development of SoSnRD. Section III introduces the principles of our approach with a particular emphasis on the 'asset' concept. Section IV details the proposed assistance mechanism using an example and finally we discuss related work in section VI before concluding in section VI.

II. Problem statement

In Emergency Response System of Systems (ERSoS), a specialized operator is responsible for deploying in a very short delay emergency teams such as fire-fighting, ambulances and police, coordinating them with hospitals or local authorities using, for instance, cartography or specific communication systems. A complete illustration was presented in the European COMPASS project [20].

The next generation of ERSoS will be implemented with Internet of Things (IoT) systems, such as Unmanned Autonomous Vehicles (UAVs) or smart medical sensors. They will integrate more information systems, allowing on-site medical teams, for example, to edit digital pre-reports or to access distant personal medical files. As the digitization of such systems is becoming more and more important, one has to deal with cyber-security concerns [17], [9], [8].

From such SoSs, we know that we need to design an operational SoS in a short-term manner: a few hours (accidents) up to a few days (military operations). Thus, we do not have enough time to follow a 'traditional' SoS development approach. The development is therefore entrusted to a unique stakeholder. Based on his/her experiences, this 'domain expert', has a perfect mastery of the constituent systems to be mobilized. Such an expert proceeds most often by instantiating and adapting a generic and proven solution to a particular context. So *the first challenge is to provide him/her with an assistance enabling him/her to: i) capitalize his/her experiences, ii) reuse this knowledge to quickly produce an architecture*. Getting the security requirements right at the early stages is essential, as they strongly impact on the architecture of the

proposed solution. Unfortunately, in most cases, the domain expert has no skill in cyber-security. He/she cannot rely on existing secure system development methodologies such as Microsoft SDL [13], OWASP [21], Secure i* [10] or SysML-Sec [3], as they are not always well adapted to SoS. There are some specific methodologies in securing SoS, such as SoSSec [12] or Security Framework Architecture [22], but these methodologies take a long time and require the strong security skill. Thus, the domain expert must interact with security experts but such a collaboration is not always possible because security experts are not necessarily available in an emergency. It is not realistic to expect that this expert, who has already made the effort of continually keeping his/her field of expertise fresh and up-to-date, becomes in addition a security expert. As the domain expert needs immediate and continuous feedback to assess the impact of his/her choices on the expected level of security, *defining a security-oriented assistance during the design phase is the second challenge and the purpose of this paper*.

III. Approach

We begin by giving some information on how we handle the solution to the first challenge mentioned above. This is necessary to understand the place, in the global development schema we propose, of the security-oriented assistance (addressing the second challenge) that this paper deals with. We then introduce the objective of this kind of assistance. We highlight the importance of the 'asset' concept, which is the core of the assistance we define. We conclude by presenting the main principles of the assistance that we propose.

A. Rapid development of SoS

To tackle the first challenge, we propose to: i) promote the reutilization, ii) automate all the development tasks as much as possible.

To facilitate the reutilization, we introduce two techniques: Domain Specific Modeling Languages (DSML) and 'generic' models. This DSML must allow to describe abstract as well as concrete architectures. The experts should be able to capitalize their knowledge in the form of reusable architecture models. To do this, the models should be more or less abstract and some of them should be generic. We advocate two complementary forms of genericity in the models: i) feature diagrams to describe domain 'reference architectures', ii) (domain) architectural patterns. As it is not conceivable to define from scratch as many DSMLs (and tools) as domains, we propose to use techniques from the Model Driven Engineering literature. We define a modeling language in which we specialize as many 'profiles' as domains. In this way, we derive a family of languages from this base language. We are currently working on this language. It is based on the concept of 'role' (an abstract or concrete constituent system such as human, software or materiel) offering some 'capabilities' (services) to other roles when they are related by a 'collaboration' relationship [7].

To automate the development, we will define a software tool dedicated to each domain expert. These tools are actually the result of extending the tool built to support the base language. The extension consists in: i) adapting the base language to the expert's domain (e.g. using stereotypes), ii) choosing a concrete syntax adapted to the domain (e.g. associating a graphical representation to a stereotype). The obtained tool will allow the expert to: edit, store, search, reuse, generalize (create an architectural pattern or a reference architecture), adapt or specialize (apply a 'pattern' or configure a reference architecture) models conforming to this DSML. A role library is also maintained by the expert. Roles are organized in a hierarchy that represents 'is-a-type-of' relationship. The expert draws from this library to design the models. Domain experts develop their architecture models by adapting existing and generic models to each context. They can gradually refine these models and make coexist more and less abstract constituent systems (roles) within the same model. We will not give here more details on this ongoing work. In the following, we make the assumption that the domain expert uses such a DSML.

B. Secure SoSnRD

To tackle the second challenge, we propose to define an assistance supported by a tool to help the domain expert assess and manage security requirements during the architecture design process. This assistance makes the assumption that the experts have little security skill. They must, however, at least be able to: i) distinguish the elements they want to protect in the architecture and ii) be aware of the quality properties relating to security: confidentiality, integrity, availability, authenticity, non-repudiation, reliability and accountability.

Based on this information and a security knowledge base, our tool uses risk analysis and provides security advice. This assistance has three levels of advice: 1) raise alerts (a list of detected vulnerabilities and associated risks), 2) propose local changes in the architecture to limit risks (configuration or replacement of certain constituent systems) or 3) global changes (application of security patterns). The expert may request at any time the launch of the assistance. However, the level of details and relevance of the advice will be better if the model is concrete and therefore will make use of real constituent systems, whose vulnerabilities are cataloged in the security knowledge base. Therefore, the process will be iterative, allowing the expert to integrate feedback from the assistance.

C. Assets that bridge the gap between domain experts and security concerns

A traditional way of analysing security risks begins by listing all assets of an organization, then identifies threats, vulnerabilities exploitable by these threats and finally estimates risks. Several definitions of 'asset' exist in literature [18], [11], [2]. ISO/IEC defines that an asset is anything that has value to the organization. In our vision, an asset is: 'anything that a domain or security expert wants to protect against a potential attack'. Domain experts and security experts have different perspectives or visions on assets.

Domain experts are willing to protect financial properties, preserve lives, confidential data or reputation of their organizations. We call these 'business assets' because they are valuable for domain experts. They should be able to express security requirements on business assets. As they are not security experts, they don't know how exactly to design the architectures to protect these assets. These business assets are the starting point for any security policy. They also delimit the competence boundary of the domain expert.

Security experts take the business assets' protections as security requirements. Their knowledge of the attack mechanisms allows them to identify other architectural elements that play key roles in obtaining the desired security level. We call these elements the 'supporting assets', such as file, database, server, etc. The security expert relies on the knowledge called 'causal chains': typical attack propagation paths in the architecture model, described using for example attack graphs. For example, the domain expert wants that a patient's health data must be confidential. An attacker can have more than one way to get this information. Each way involves an asset causal chain. One possible attack scenario is: attackers remotely connect to the network via malware. The malware will then spread, sometimes even directly from a physical access point, until it reaches the workstation where a database that stores personal information is installed. The corresponding causal chain is: information ← database ← protection mechanism ← workstation ← network ← physical access point (final goal: patient's data).

One of the key issues in providing a security assistance is being able to deduce from the business assets, the supporting assets and the security properties that they must respect.

D. Global vision of the security-oriented assistance

We show our vision of bridging the gap between domain expert and security concerns in Fig. 1. The horizontal axis illustrates three worlds: domain, asset and security. The asset world is the connection between the domain world and the security world. The vertical axis demonstrates three different abstraction level: requirement, design and implementation. The blue part indicates the vision from domain expert while the green part displays the security concerns. In our vision, we promote three types of asset: business, pivot and supporting asset. Pivot assets are introduced to facilitate the transition between business assets and supporting assets.

Business asset, which the domain expert wants to protect, is tagged by the domain expert on the architecture model. It is the final goal from the point of view of attackers and the consequence of an attack from the point of view of defenders.

Pivot asset is the core of an attack, which means no matter which kind of mechanism/platform that an attacker exploits, it is the pivot asset inside of the platform that is the sub-goal of an attacker to achieve the final goal (business asset) or the core of an attack. Pivot assets are independent of the domains. For example, a confidential military plan stored in a database, here the confidential information is our pivot asset but not the database. Example of pivot assets are resource, identity,

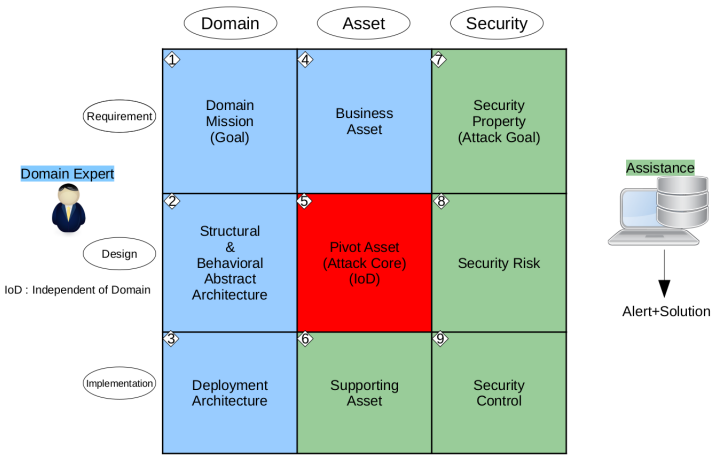

Fig. 1. Global Vision

function, communication, access, etc. They belong to three families: *data*, *process*, and *human related asset*. Pivot assets are fine-grained assets that we need to assess their security needs or sensitivity, but not their vulnerabilities, because no vulnerability exists there. Once a business asset is tagged, we deduce the pivot assets needing protections to avoid the compromise of the corresponding business asset.

Supporting assets provide supports for pivot assets. We'll assess their vulnerabilities, but not their security needs or sensitivity, because vulnerabilities exist inside, but without the pivot assets they support, they are not valuable to the organization. In the above example, the database is a supporting asset which supports the pivot asset 'information'. A successful attack is reachable only from attacking the combination of a pivot asset and a supporting asset, lacking anyone of them can not lead to a successful attack.

The assistance is based on a knowledge base comprising: i) a mapping table to identify pivot assets, ii) a mapping table associating pivot assets with security risks, iii) a database describing a list of known vulnerabilities for each constituent system (if it is a concrete system) or (if an abstract one) vulnerability categories, iv) causal chains describing possible attack paths. This knowledge base is created and maintained by a security expert independently of the domain expert's activities.

IV. Security Assistance in Action

In this section, we present how the security assistance works through a case study. We proceed step by step and present: i) how the assistance interacts with the user and ii) the mechanisms implemented to achieve these results.

A. Case study and development context

We take an example of a mission of rescuing an injured person. In order to respond to this situation, a domain expert has to deploy an emergency vehicle equipped with a connected heart monitor. The injured person needs to be carried to an hospital. His/her heart must be checked and data must be

automatically uploaded to his/her personal medical file. The file is accessible from the hospital center.

The domain expert designs this SoSnRD using a specific DSML. The first version of the architecture model is perhaps the result of an adaptation of an architecture feature diagram applicable to this type of mission. The expert searches for such a template in the model database. Once found, the tool allowed him/her to configure this template model to meet the particular context of the mission. This adaptation imposes the determination of all the variation points presented in the template. The determination, in this case, can be done by answering questions dealing with: the size (how many people to repatriate), the severity of injuries, the distance and the accessibility of the emergency area (mountainous area, plain, etc.), the risk level of the sector (the presence of enemy forces for example). It is very likely that this first version of the architecture contains many abstract constituent systems.

The expert can, at any design period, requests the security assistance. He/she can get some fairly generic information and advice when most of the constituent systems are still abstract. We will present the assistance process in a more interesting scenario, where the expert requests the launch of the assistance on a more concrete version of the architecture. Thus, we suppose that the domain expert has improved the initial model to integrate some concrete elements as shown in Fig. 2.

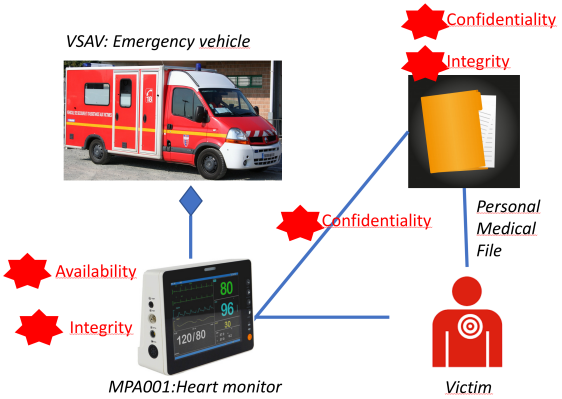

Fig. 2. Tagged Deployment Architecture Model

B. Process of the security assistance

In Fig. 3 we illustrate the assistance process using the SysML activity diagram. In this figure we show the actions performed by the user and those by the assistance. The input and output information of each activity are also described. We will now examine each step of this process, when the assistance is launched on the model in Fig. 2.

C. Identifying business assets

The assistance asks the domain expert to tag security properties on the model elements having security value for him/her. These tagged elements are called 'business assets'. He/she can choose at each launch to delete already tagged elements or to add others. If a tagged model element is an

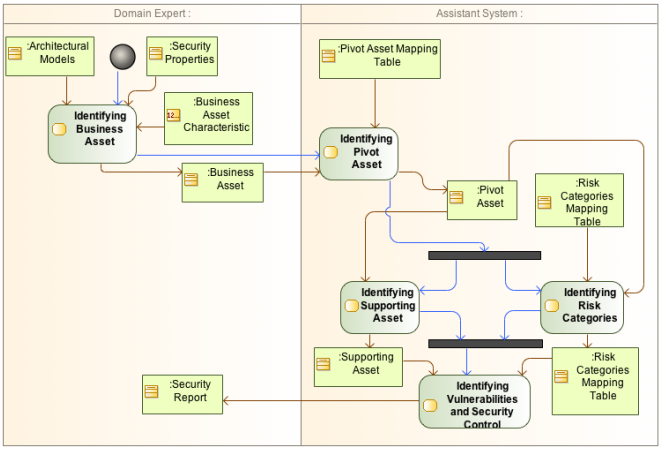

Fig. 3. Illustrated Process of the Security Assistance

TABLE I
PIVOT ASSET MAPPING

Security Property \ Category	Data	Process	Human Related Asset
Confidentiality	Information Credential	Communication	Identity Activity Trace
Integrity	Information Credential Identifier	Resource Function Service	Identity Privilege
Availability	Information	Resource Function Service Access Communication AllocableMemory	-
Authenticity	-	Trust Access	Identity Trust
Non-repudiation	Identity Identifier	Trust Access	Identity Trust
Reliability	Information	Function Service Control	Proper Usage Trust
Accountability	-	Trust	Identity

abstract one, the tag is also maintained during design until the model element becomes a concrete one. Then, the assistance asks the expert to classify the tagged element into one of the three asset categories: data, process and human-related asset according to the security properties tagged on it.

In Fig. 2, the expert has identified 3 business assets: *PersonalMedicalFile* (type:*data*) associated with the security properties *confidentiality* and *integrity*, *sendsData* (type:*data*) associated with *confidentiality* and *ConnectedHeartMonitor* (type:*process*) associated with *integrity* and *availability*.

D. Identifying pivot assets

The assistance uses Table I to identify pivot assets. This table is a mapping between pairs of (asset categories, security properties) and pivot assets. This table allows for each pair to get automatically a list of pivot assets. In this case study, for the business asset *PersonalMedicalFile*, the pair *(data,*

confidentiality) allows to identify the following pivot assets: *information* and *credential*. For the *monitor* business asset and its pair *(process, availability)*, we identify: *resource*, *function*, *service*, *access*, *communication* and *allocable memory*. Applying the same process on each business asset conducts to identifying all pivot assets.

E. Identifying supporting assets

For each pivot asset identified, the assistance tries to identify a set of 'supporting assets' for further vulnerability identification. The objective of this step is to attach the pivot asset to an architectural element. In order to identify the corresponding supporting assets, the assistance uses business assets. Indeed, high level architectural elements, on which business assets were set, are refined towards concrete architectural elements. Traceability mechanisms such as SysML allow the assistance to find them. As mentioned above, the domain expert can start the assistance at any time. Thus architectural elements could be more or less concrete. Depending on the architecture's maturity, following situations are possible:

In the first situation, the abstract architectural element on which the business asset was set is refined by a constituent system or a combination of constituent systems. In this case, the assistance identifies the constituent system (or all elements of the combination) as the supporting asset. For instance, concerning the pivot asset *function*, the assistance identifies a constituent system called *MPA001 heart monitor*. It is composed of sub-systems. One of them is a monitor device Siemens 22 TFT. The later is a supporting asset which supports the pivot asset *function*.

In the second situation, the business asset was set on an abstract architectural element without clearly identifying the capability. Thus, the assistant generates a message to the domain expert saying that the security analysis is not yet possible on this asset. The assistance process would be started another time later. For instance, the assistance detects that *PersonalMedicalFile* is typed as *Data*. A *Data* must be supported by an architectural element, forgotten by the domain expert. Thus the assistance reports the problem to the latter.

In the third situation, no concrete element is defined to realize an abstract element but the abstract element is identified as a generic role attached to an identified capability (a *server* for instance). In this case, the remainder of the process will provide generic recommendation concerning all server roles. For instance, let's consider that the domain expert made a correction on the architectural model. He/she has created two abstract roles: *hospital* and *server*. *Server* manages the *PersonalMedicalFile*. He/she doesn't know which type of server would be deployed. Nevertheless, *server* is a supporting asset. We can then provide a vulnerability category corresponding to this supporting asset.

F. Identifying Risk categories

This activity is performed in parallel with the previous one. For each pivot asset identified, the assistance will search in Table II the corresponding risk categories. Only a small part of the designed table is shown. The risk categories are found in CVE[1] and CAPEC[2] databases. CVE provides a reference method for publicly known security vulnerabilities and exposures. CAPEC helps by providing a comprehensive dictionary of known patterns of attack employed by adversaries to exploit known weaknesses in cyber-enabled capabilities. In the tool database, we map each pivot asset to the risk categories that may impact on it. The output of this step is some special risk categories identified. For concrete constituent systems, precise vulnerabilities are identified. For abstract constituent systems, more high level vulnerabilities are identified. In the example, *DOS*, *code execution* and *overflow* were identified for the *function* pivot asset and gain *information* and *gain privilege* were identified for the *information* one.

TABLE II
RISK CATEGORIES MAPPING WITH PIVOT ASSETS

Pivot Asset \ Risk Category	DOS	Code Execution	Overflow	Gain Information	Gain Privilege	...
Function	✓	✓	✓			
Information				✓	✓	
...						

G. Identifying vulnerabilities and security controls

For each supporting asset identified, the assistance searches in the constituents systems database the detailed vulnerabilities exploitable in these supporting assets according to each risk category. For each vulnerability identified, it proposes a list of security controls to decrease the security risk caused by these vulnerabilities.

For instance, concerning the *Siemens 22 TFT* supporting asset and the *function* pivot asset, a precise vulnerability is identified in relation with the *DOS* risk category : *CVE-2015-2177 Siemens S7-300 CPU Denial-of-Service Vulnerability*. Concerning the *server* supporting asset and the *information* pivot asset, a general vulnerability is found concerning the *gain information* risk category : *CVE-2016-3298: Internet Explorer Information Disclosure Vulnerability, the vulnerable Systems can be Microsoft Internet Explorer 9, 10, 11, Microsoft Windows 7, Windows Server 2008 and Vista*.

For each identified vulnerability, *security controls* are proposed to decrease risks and report alerts. For instance, concerning the vulnerability CVE-2015-2177, the assistance proposes the following security control mechanisms: (i) *applying protection-level 3 (Read/Write protection)*; (ii) *applying the cell protection concept*; (iii) *using VPN for protecting network communication between cells*; (iv) and *applying a Defense-in-Depth strategy*. For the vulnerability CVE-2016-3298, the assistance proposes the following security control: *patching*.

The assistance in this example provides a list of alerts (assistance of level 1) and proposals of security controls

[1]CVE: https://www.cvedetails.com/vulnerabilities-by-types.php
[2]CAPEC: http://capec.mitre.org/

(assistance of level 2) related to the identified vulnerabilities. Applying these security controls would not be easy for the domain expert. Indeed, in the SoS domain, a constituent system exists "as is". In the best case, the system can be parameterized using specific security interfaces. In the worst situation, there is no such possibility and no other possible substituting systems. In this case, an alert is sent to the domain expert informing him/her of the vulnerabilities and the risks.

V. Related work

Emergency (Crisis/Disaster) Response (Management) SoS (ERSoS) are complex social-technical entities that handle relief and recovery operations in disaster situations [14], [6], [4], [5]. A disaster is a continuously unfolding situation, marked by changes in urgency, scope, impact, the types of appropriate responders, and the responders' needs for information and communication [14]. Examples include epidemic/disease outbreaks, public health emergency, medical resource management, bio-terror attacks, network-centric defense forces, floods, bushfires, fire-fighting and police [14], [4], [1], [5]. The SoSnRD is a superset of ERSoS. Moreover, such a wide span poses security issues, such as establishing trust between the involved actors, establishing, enforcing and monitoring security policies and ensuring data privacy [16]. ERSoS security solutions need to be flexible [16]. Not many security solutions for ERSoS have been proposed. Among these, [4] proposes a centralized client-server identity management (identity provision and preservation, authorisation and authentication) solution. However, this is limited to identity management, not addressing other security issues and is not decentralized, while we need a central decision-making entity to resolve possible conflicts in access rights. Our work, which deals with assets, security vulnerabilities and controls, is therefore a first stepping stone to address these issues.

VI. Conclusion

In this paper, we have introduced a special kind of system of systems: system of systems needing rapid deployment which are sometimes deployed in an environment where security is an important concern. We have shown that their developments require specific methods and tools. These developments are led by domain experts who, in most cases, have little security skill. Thus, we have presented the principle of an assistance able to derive automatically security alerts and controls from a list of (business) assets specified by domain experts. We have identified a possible process and verified it with a case study. We have identified the actions the domain expert must conduct and those executed by the future security assistant tool. We have defined a knowledge base supporting the first and second level of the assistance taking the form of correspondence tables. We plan now to use model-driven engineering techniques such as model transformations to develop a first version of this assistant tool. We will firstly focus on providing an assistance of level 1 (raise alerts) and 2 (propose local changes). In the future, the third level of the assistance (propose global changes) is needed to be tackled.

Acknowledgment

This work is funded by the DGA[3] and the PEC[4]

References

[1] E. Akdogan, Y. Peres, and A. Yaeli. A system of systems approach in responding to events and addressing public health needs. In *Proc. of the 4th Int. Conf. on Risk Analysis and Crisis Response*, Istanbul, 2013.

[2] ANSSI. Classification method and key measures, 2014.

[3] L. Apvrille and Y. Roudier. *Model-Driven Engineering and Software Development*, chapter Designing Safe and Secure Embedded and Cyber-Physical Systems with SysML-Sec, pages 293–308. Springer, 2016.

[4] A. Arabo, M. Kennedy, Q. Shi, M. Merabti, D. Llewellyn-Jones, and K. Kifayat. Identity management in system-of-systems crisis management situation. In *6th Int. Conf. on System of Systems Engineering*, pages 37–42, 2011.

[5] G. Beydoun, S. Dascalu, D. Dominey-Howes, and A. Sheehan. Disaster management and information systems: Insights to emerging challenges. *Information Systems Frontiers*, 20(4):649–652, Aug 2018.

[6] S. Chandana and H. Leung. A system of systems approach to disaster management. *IEEE Communications Magazine*, 48(3):138–145, 2010.

[7] Imane Cherfa, Salah Sadou, Nicolas Belloir, Régis Fleurquin, and Djamal Bennouar. Involving the application domain expert in the construction of systems of systems. In *13th Annual Conference on System of Systems Engineering, SoSE 2018, Paris, France, June 19-22, 2018*, pages 335–342, 2018.

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

[9] H. Dogan, C. Ncube, S. L. Lim, M. Henshaw, C. Siemieniuch, M. Sinclair, V. Barot, S. Henson, M. Jamshidi, and D. DeLaurentis. Economic and societal significance of the systems of systems research agenda. In *Int. Conf. on Systems, Man, and Cybernetics*, pages 1715–1720, 2013.

[10] G. Elahi and E. Yu. A goal oriented approach for modeling and analyzing security trade-offs. In *Proc. of the 26th Int. Conf. on Conceptual Modeling*, pages 375–390, 2007.

[11] European Union Agency for Network and Information Security. Threat and risk management - glossary. https://www.enisa.europa.eu/topics/threat-risk-management/risk-management/current-risk/risk-management-inventory/glossary.

[12] J. E. Hachem, Z. Y. Pang, V. Chiprianov, A. Babar, and P. Aniorte. Model driven software security architecture of systems-of-systems. In *23rd Asia-Pacific Software Engineering Conf.*, pages 89–96, 2016.

[13] M. Howard and S. Lipner. *The Security Development Lifecycle*. Microsoft Press, Redmond, WA, USA, 2006.

[14] Marijn Janssen, JinKyu Lee, Nitesh Bharosa, and Anthony Cresswell. Advances in multi-agency disaster management: Key elements in disaster research. *Information Systems Frontiers*, 12(1):1–7, Mar 2010.

[15] B. Josko. Designing for adaptability and evolution in system of systems engineering, 2015.

[16] I. H. Krueger, M. Meisinger, M. Menarini, and S. Pasco. Rapid systems of systems integration - combining an architecture-centric approach with enterprise service bus infrastructure. In *IEEE Int. Conf. on Information Reuse Integration*, pages 51–56, 2006.

[17] M. Merabti, M. Kennedy, and W. Hurst. Critical infrastructure protection: A 21st century challenge. pages 1 – 6, 2011.

[18] NIST. Specification for asset identification 1.1. Technical Report NISTIR 7693, June 2011.

[19] North Atlantic Treaty Organization. Nato architecture framework, version 4, 2018.

[20] COMPASS Project. Accident response use case engineering analysis report using current methods & tools. Technical Report D41.1, COMPASS Project, www.compass-research.eu/Project/Deliverables/D411.pdf, 2013.

[21] The OWASP Foundation. OWASP Secure Software Development Lifecycle Project, 2017.

[22] D. Trivellato, N. Zannone, and S. Etalle. A security framework for systems of systems. In *2011 IEEE International Symposium on Policies for Distributed Systems and Networks*, pages 182–183, June 2011.

[3] DGA:https://www.defense.gouv.fr/dga

[4] Pole d'Excellence Cyber : https://www.pole-excellence-cyber.org/