Investigating Attack Propagation in a SoS via a Service Decomposition

Elena Lisova
Mälardalen University
Västerås, Sweden
elena.lisova@mdh.se

Jamal El Hachem
Université Pau & Pays Adour, LIUPPA
Pau, France
jamal.elhachem@univ-pau.fr

Aida Čaušević
Mälardalen University
Västerås, Sweden
aida.causevic@mdh.se

***Abstract*—A term systems of systems (SoS) refers to a setup in which a number of independent systems collaborate to create a value that each of them is unable to achieve independently. Complexity of a SoS structure is higher compared to its constitute systems that brings challenges in analyzing its critical properties such as security.**

An SoS can be seen as a set of connected systems or services that needs to be adequately protected. Communication between such systems or services can be considered as a service itself, and it is the paramount for establishment of a SoS as it enables connections, dependencies, and a cooperation. Given that reliable and predictable communication contributes directly to a correct functioning of an SoS, communication as a service is one of the main assets to consider. Protecting it from malicious adversaries should be one of the highest priorities within SoS design and operation. This study aims to investigate the attack propagation problem in terms of service-guarantees through the decomposition into sub-services enriched with preconditions and postconditions at the service levels. Such analysis is required as a prerequisite for an efficient SoS risk assessment at the design stage of the SoS development life cycle to protect it from possibly high impact attacks capable of affecting safety of systems and humans using the system.

I. Introduction

Recent advances in technologies allow building more complex and interconnected systems even for safety-critical applications. Such systems ultimately present a collection of systems that share their capabilities and resources to achieve new functionalities and increase the overall efficiency compared to traditional systems, defined as Systems of Systems (SoS). Systems that are a part of an SoS are called Constituent Systems (CS). Examples of SoS spread starting from the Internet of Things (IoT) applications and up to critical cooperating systems like car platoons.

Advantages of SoS in terms of their capabilities and performance, are making them an appealing choice for safety-critical applications as well. Such systems are highly connected and consequently, security becomes a paramount to address and support safety guarantees, as a connected system is not safe if it is not secure [1]. Therefore, besides the challenge of engineering nature, taking into account the complexity of the communications between CSs or services, an additional challenge is the consideration and analysis of important quality attributes such as security. SoS security has gained growing attention during the recent years [2]. However, in order to be able to successfully engineer an SoS, security needs to be given even a higher priority and new, more structured analysis approaches are needed to deal with this challenge.

Over the last years, there has been a growing awareness of SoS being exposed to severe security attacks. Their large scale infrastructure, complexity of their architectures and a large interaction surface makes them extremely vulnerable to malicious attacks. These attacks could be established by a single vulnerability or a sequence of several triggered vulnerabilities induced by the SoS communications and connected in an unknown way resulting in hazardous situations/emergent behaviours. The challenge of SoS security analysis lies in its complexity and interdependencies between its CS, as such attacks cannot be analyzed by evaluating CS and services independently. It requires the assessment of the complete SoS including all CS and services as well as their connections with focus on possible attacks in order to identify possible hazardous situations. In this work we investigate the effect of attack propagation in terms of service-guarantees.

To investigate attacks propagation in SoS, we consider an SoS representation using services, where an attack can affect a service precondition and be propagated further to other services by the affected service postcondition being not fulfilled. The contribution of this work is in mapping between affected service postconditions and preconditions of other services within the SoS. Such analysis allows to evaluate an effect of attacks on SoS and is a prerequisite to an efficient SoS risk assessment.

To illustrate the approach we use an example of autonomous quarry[1], where we consider communication between its CS being a service itself and analyze propagation of an attack targeting a communication service within the quarry. Such SoS is critical due to people being in the loop, the cost of its constituent systems and the operational process being disrupted. The focus of this work is on communication between CS of SoS, considered as a service, and possibly facilitating attack propagation. Given such setting, this use case with several types of communication being part of the communication service and with CS relaying on the communication, perfectly illustrates the need for such a propagation analysis as a prerequisite for the efficient and correct risk assessment.

[1]The use case is inspired by the real-world use case presented by Volvo https://www.volvoce.com/global/en/this-is-volvo-ce/what-we-believe-in/innovation/electric-site/

The rest of the paper is organized as follows. Section II presents necessary background regarding a service and security terminology, whereas Section III describes the use case and its representation via services. Next, the attack propagation and the mapping as a prerequisite for an SoS security analysis are considered in Section IV. Furthermore, related work is presented in Section V, and finally Section VI concludes the paper.

II. Background

A. Services

A *service* can be seen as a set of functions provided by a service provider to a service user, accessible through an application programming interface [3]. Services can be created, invoked, composed and destroyed on demand. They are platform independent and applicable to heterogeneous applications. A composite services contains two or more services with the main goal of a reusable functionality being provided by existing services in a low cost and rapid development process on demand. One of the main characteristics of services is separation of interfaces from the service behavioral description. A publicly available service interface is visible to service users and used to find and invoke services most suitable for their needs. Service interfaces are as well used to provide the information regarding service pre-, and postconditions that are predicates that constrain the start of the service execution and provide the output guarantees that must hold after a service execution, respectively. On the other hand, internal behavior-related is hidden from a service user and available only to service developers. In such a context, a service becomes a single point of maintenance for a common functionality.

A service composition can be achieved either through *orchestration* that includes a central controller responsible for scheduling service execution according to the user demands or *choreography* which enforces a mechanism of message exchange between participants in the composition, without requiring a central coordinator. A choreography relies on the fact that each service in composition has knowledge when to execute its operations and with whom to interact.

B. Security

In this work, we consider security as a service property provided within the SoS. A *security attack* is an attempt to destroy, expose, alter, disable, steal or gain unauthorized access to or make unauthorized use of a service [4]. In the context of SoS, an attack could result from the exploitation of one or several vulnerabilities that menaces one or several services and are triggered through the communications/interactions between these services.

Each attack could be activated by one or several preconditions and results in one or many postconditions. The latter conditions could be defined by a security expert or it could be extracted from vulnerability databases, such as the Common Vulnerabilities and Exposures (CVE) catalog[2], which offers a standardized description allowing the extraction of several information such as the attack pre-, and postconditions. For example, for an SQL injection attack that consists of a code injection to insert and run nefarious SQL statements on a database, the preconditions are the following: “user input is incorrectly filtered for string literal escape characters embedded in the SQL statements” or “user input is not strongly typed and unexpectedly executed”; and the postconditions are “information disclosure” and / or “total shutdown of the affected resource”.

[2]https://cve.mitre.org/

III. Use Case

In the following we describe our use case that is an autonomous quarry, in terms of an SoS including the overview of the most important service in such environment.

A. Autonomous Quarry

An autonomous quarry consists of the following CS: *(i)* a remote control room, where an operator has a constant overview of all processes and a possibility to take over control if needed; *(ii)* a fleet of autonomous carriers, that carry the load, e.g., stones of different granularity; *(iii)* a charging station for carriers; *(iv)* a point with stone extraction, where a wheel loader loads the carrier with higher granularity stones; *(v)* a point where large stones are crashed into smaller ones; *(vi)* a factory/storage facility where carrier delivers lower granularity stones. The intelligence, i.e., decision making is placed in both the control room and locally at carriers. Generally in a normal operation mode, the control room communicates with every carrier, it sends an updated map of the quarry (i.e., routes, other carriers locations and statuses), current task for the carrier, its direction to go, acceleration, and speed. Thus, in the normal operation mode, the carrier just follows the receiving commands.

However, as a quarry has harsh environmental conditions (e.g., dust), it is possible that quality of communication with the control room degrades. Hence, a carrier also has sensors to locate objects near by, their speed and is able to make its own decision about required action even if it overrules the command from the control room, e.g., an emergency stop when an object detected in a dangerously close proximity. To overrule the prescribed action and switch to a self-control mode, a carrier has to make a decision about communication channel with the control room being unreliable enough and detect a condition requiring a emergent response. After the response the carrier has to execute one of the failing safe scenarios. The remote control room also has a possibility to broadcast an emergency stop message that is a safety measure to react upon detected hazardous situation. Due to the harsh environment in the quarry, additional techniques like relaying can be used to increase the reliability of a broadcast communication to an acceptable level. In this work, we do not go into details about the physical level of communication in the quarry, as we are more interested in the logical connections at higher levels.

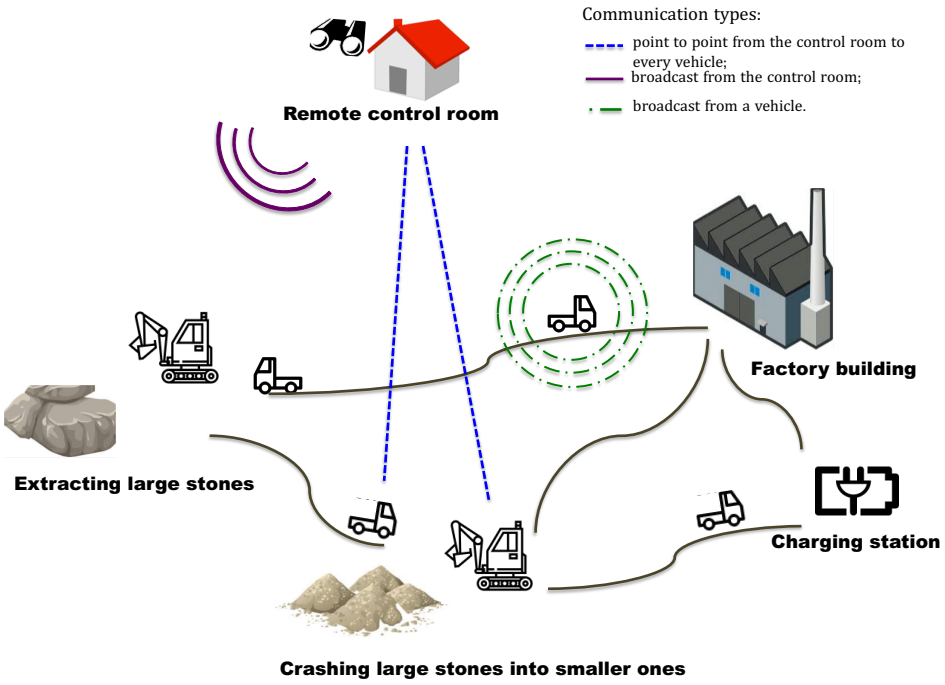


Fig. 1: Types of communication in the considered autonomous quarry.

As demonstrated in Fig. 1, we assume the following types of communication to exist in the autonomous quarry:

(*i*) point-to-point (p2p) communication between the control room and every vehicle in the quarry, it carries control information about vehicle's actions and relevant information about current status of other vehicles and quarry in general;

(*ii*) broadcast communication from the control center with the main function to propagate the emergency stop message to all vehicles if needed;

(*iii*) broadcast communication from each vehicle with a smaller range than Type (*ii*) communication, as it's main functionally is to broadcast a status information regarding vehicles in the quarry to increase awareness of the nearby vehicles. However, the functionality is redundant in the normal operational mode.

We do not go deep into details about how communication is realized, however we assume that the IEEE 802.11p standard is used. 802.11p belongs to the 802.11 stack but is focused on vehicle communication and targets establishing Wireless Access in Vehicular Environments (WAVE), it includes support of data exchange between high-speed vehicles and between the vehicles and the roadside infrastructure.

B. Quarry as SoS Composed of Services

In this work, we consider a quarry as a SoS, composed of a set of services, where we distinguish between: *a control level service*, *a communication level service* (S_{com}) and *a system level service* i.e., functional services, as depicted in Fig. 2. At the control level, we assume that the control room is in charge for the decision making process in quarry, as well for general information regarding possible operational changes. A communication service is a composite service that enables different types of communication within the quarry. Table I provides a description of communication types (subservices in a composite service) provided in this use case, together with their pre-, and postcontions, ($precond\_S_{comN}$) and ($postcond\_S_{comN}$), respectively. In order to enable any of these communication types, their respective precondition must hold before their execution, as well as postcondition has to be guaranteed after the service execution. At the bottom level we assume system level services that represent the behavior of autonomous vehicles.

IV. Attack Propagation in Terms of Service-Guarantees

In this section we describe some possible attack scenarios for our use case and analyze how these attacks might be propagated through the system by mapping attacks and services within an SoS.

A. Possible attack scenarios in autonomous quarry

In this work, we investigate two types of attacks: *(i)* attacks on communication services resulting in the propagation of false information to autonomous vehicles and a vehicle reaching hazardous state (i.e., a man-in-the-middle attack in p2p communication), and *(ii)* attacks on external devices attached to the autonomous vehicle (e.g., sensors, radars, lidars, etc.) resulting in a vehicle receiving incorrect information about surrounding environment leading to the potential hazardous state (i.e., blinding, jamming, relay, and spoofing attacks).

We suggest some possible realistic attacks inspired from our study of the autonomous vehicles related work and attack news

TABLE I: Communication service decomposition into sub-services.

S_{com1}: point-to-point communication between the control room and every vehicle in the quarry (a normal operation mode).	
$precond\_S_{com1}$	– a wireless data link between two points exists.
$postcond\_S_{com1}$	– timely delivery of control information to vehicles; – timely delivery of status information to the control room; – wireless data link between two points exists.– the normal operational mode active.
S_{com2}: broadcast communication from the control room (an emergency mode).	
$precond\_S_{com2}$	– an attack to the quarry infrastructure has been detected.
$postcond\_S_{com2}$	– the emergency stop message delivered as soon as the attack has been detected; – activation of the mechanisms supporting the emergency stop message propagation; – the emergency mode active.
S_{com3}: broadcast communication from a vehicle.	
$precond\_S_{com3}$	– all sensors, radars and lidars in normal operation mode.
$postcond\_S_{com3}$	– timely delivery of awareness information to the surrounding vehicles.

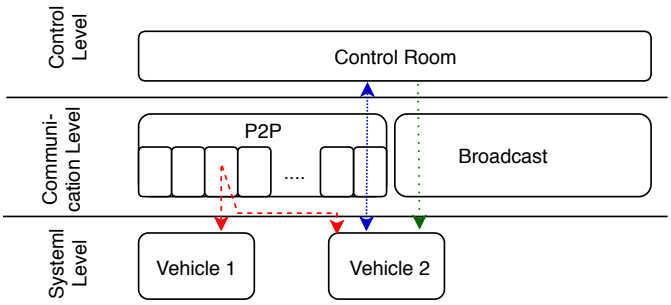


Fig. 2: An autonomous quarry as SoS.

such as those in [5], [6], [7], [8]. Table II synthesizes some of the potential attacks and their pre-, and post-conditions.

B. Mapping attack postconditions and service preconditions

To investigate the attack propagation problem in terms of service guarantees, we propose to describe an attack in a services fashion, i.e., by defining its pre-, and postconditions. Thus, if an attack postconditions and a service precondition are described by means of the same language, we can map them to demonstrate that a service is affected by the attack. Fig. 3 shows an example, when $postcond\_A_1$ maps to $precond\_S_2$. The red thread in Fig. 3 illustrates how the attack propagates within the SoS: due to $Attack_1$ being executed, $precond\_S_2$ is not fulfilled anymore and thus $postcond\_S_2$ does not hold, thus there is no guarantees on correct execution of the service $Service_2$; in its turn $precond\_S_N$ is not either fulfilled anymore, as it requires $postcond\_S_2$ to hold for it (e.g., in case of a serial service composition), which consequently leads to the SoS output not being delivered correctly. A service can be potentially composed of several services and in this case possible further propagation has to be considered as demonstrated in Fig. 3 (b).

Let us assume that the attack A_1 is executed within the quarry. In that case the control level services (S_{com2}) lose their capabilities to send correct and timely information to the vehicles in the quarry. This can result in a situation where it is impossible to send a correct and timely information regarding the work to be done, status of the quarry, appropriate speed, etc., unless some other redundant communication channel exists (i.e., cellular network). Moreover, if control services fail to detect such an attack, a false information might be sent to autonomous vehicles resulting in hazardous events (i.e., damage of equipment, or in the worst case death of people, e.g., a wheel loader driver). In the best case, such an attack will be detected and emergency stop will be issued, resulting in a safe state, but loss of time and additional cost due to the unplanned quarry halt.

On the other hand, assuming attacks A_2 or A_3 that tamper with sensors and cameras attached on the autonomous vehicle, system level services (S_{com3}) fail to get the real information about surrounding environment and possibly resulting in a collision with other vehicles, or existing obstacles in the quarry. In both cases, a quarry will end up with either unplanned stop, but equipment and people being protected, or in a potentially severe damage to both equipment and people, with additional cost and loss of time.

C. Discussion

An attack might be propagated at different levels: (*i*) an attack level, i.e., a situation when a postcondition of one attack, exploited by an adversary, satisfies vulnerability preconditions of another attack; (*ii*) a service level, i.e., a case when a postcondition of an attack affects a service precondition. This work is focused on the latter. However, for a full analysis of attack propagation within an SoS both aspects have to be considered. There are works considering attack propagation via vulnerabilities connections [9], [10], and building upon those, it is possible to extend the approach proposed in this work to cover both types of propagation as well as their combinations.

New vulnerabilities are constantly being discovered [11] and consequently new attacks constructed, thus a system design and system security analysis have to account for existence of unknown vulnerabilities. In this paper we consider only known vulnerabilities and attacks that can be propagated in many different and unknown ways. Such analysis is suitable for a system design phase or for system security evaluation. However to be applied efficiently during the run-time there has to exist a support for run-time adaptation that enables consideration of unknown vulnerabilities, as well. Providing a

TABLE II: Attacks targeting the autonomous quarry sub-services.

A_1: Attack targeting 802.11p	
$precond\_A_1$ $precond\_A_2$	– jamming or blocking a specific frequency; – mis-configurations or incomplete configurations.
$postcond\_A_1$	– prevent devices from transmitting data (blocking the communications); – modification and/or falsification of data; – lack of access to situational awareness information; – inability to stop the machine remotely or in an emergency.
A_2: Remote relay attack on LiDAR sensors.	
$precond\_A_2$	– relaying signals sent from the LiDAR sensor of the target quarry to another position.
$postcond\_A_2$	– failure in detecting or late detection of an object; – spreading fake echos (making real objects appear closer or further than their real locations); – sending fake warnings; – triggering emergency brakes.
A_3: Attack on cameras that aims to detect the traffic signs and other objects.	
$precond\_A_3$	– hitting the camera with bursts light.
$postcond\_A_3$	– overexposition of images; – hiding the objects from autonomous quarries.

support for real-time analysis of an above described case is a possible future extension of the proposed approach.

V. Related work

Recently, diverse studies in the field of SoS have been published, some of them highlighting the importance of considering security as a service when engineering SoS and the usefulness of investigating the effect of attack propagation to guarantee the SoS services.

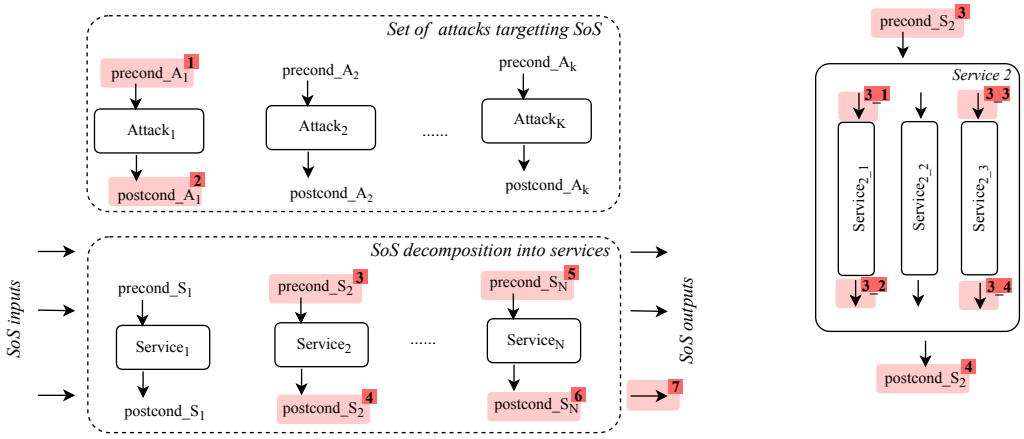
Guariniello and DeLaurentis [12] present a modified version of the Functional Dependency Network Analysis to make it applicable for SoS to analyze the internal and external impact of attacks on the SoS interdependencies. To perform the analysis, the SoS is represented as a directed network with nodes to represent the CSs or their capability and links to represent the dependencies. However in this study the security is barely considered by adding a weight indicating the availability of data to model the effect of an attack. There are no security concepts describing the vulnerabilities or the attack, neither the SoS hazardous situations.

Wang et al. describe in-vehicle network being corrupted by hacking-into-vehicle attacks [13]. Authors considered security issues from in-vehicle network and external network, - or multi-fusion. Such an attack exploits in-vehicle control area network (CAN) vulnerabilities together with flaws of on-board units (OBU). In the paper they consider both short-range attack where attackers invade in-vehicle network or send wrong control commands to in-vehicle CAN, as well as a long-range attack where an adversary, uses radio functions to compromise hardware devices, gains an access to in-vehicle CAN. Katewa et al. analyze security issues for a resource-constrained Unmanned Aerial Vehicle (UAV) [14], usually used for surveillance, reconnaissance, etc., or in military purposes. They consider attacks that compromise UAV sensor measurements, such as Global Positioning System (GPS) and a vision camera, and explore a Denial-of-Service (DoS) attack enabling an adversary to compromise received GPS signal. To mitigate effects of DoS attacks, authors rely on redundant sensors usage such as an extra camera sensor.

Attack models targeting communication in a system might be divided into attack models based on targeted functionalities in different network layers [15], or grouped based the common goal, such as DoS and deception attacks [16]. In some cases such attacks might aim for specific protocols, e.g., HTTP/2 Internet service [17], or for example jamming attacks targeting wireless networks [18]. Considering radio-frequency identification (RFID) applications several other attacks might be introduced such as: a forgery attack, a replay attack, a man-in-the-middle attack, a tracking attack [19], DoS, an eavesdropping and scanning [20] and, finally, those attacks focusing on air interfaces [21].

VI. Conclusions

Given advances in technologies allowing increased connectivity, nowadays many systems are being developed as an SoS including those in the safety-critical domain, as well. In this work we focus on communication between CS as a service, for which security is a critical property to analyze at both a service and an SoS level. The communication within SoS CS or services as well as the connections between its CS or services facilitates the surface for an attack propagation. Thus, we propose an approach to analyze such propagation by specifying CS of an SoS in terms of services, as well as annotate attacks of interest with their corresponding pre-, and postconditions. Furthermore, we consider a simplified way of mapping between them, which allows identification of services that are affected by an attack and how the attack consequences can be propagated within a SoS. Such analysis



(a) (b)

Fig. 3: Attack propagation through a SoS decomposed into services: (a) within a SoS; (b) within a service.

that incorporates possible propagation on attacks and services levels, is required for an adequate risk assessment.

In the future work, we plan to model and formalize the approach proposed in this work. Our aim is to use formal methods to enable the analysis and therefore a set of transformation rules that will cater for this, needs to be introduced. Also, we plan to consider an attack propagation that can trigger other attacks in order to incorporate more complex scenarios of attacks.

ACKNOWLEDGMENT

This work is supported by the SAFSEC-CPS project funded by KKS and the Serendipity project funded by SSF.

REFERENCES

[1] R. Bloomfield, K. Netkachova, and R. Stroud, "Security-informed safety: If it's not secure, it's not safe," in *Software Engineering for Resilient Systems*, A. Gorbenko, A. Romanovsky, and V. Kharchenko, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 17–32.

[2] P. Nguyen, S. Ali, and T. Yue, "Model-based security engineering for cyber-physical systems: A systematic mapping study," *Information and Software Technology*, vol. 83, pp. 116 – 135, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0950584916303214

[3] M. Broy, I. H. Krüger, and M. Meisinger, "A formal model of services," *ACM Trans. Softw. Eng. Methodol.*, Feb. 2007. [Online]. Available: http://doi.acm.org/10.1145/1189748.1189753

[4] International Organization for Standardization, "ISO/IEC 27000:2018 Information technology – Security techniques – Information security management systems – Overview and vocabulary," 2018.

[5] H. Shin, D. Kim, Y. Kwon, and Y. Kim, "Illusion and dazzle: Adversarial optical channel exploits against lidars for automotive applications," in *Cryptographic Hardware and Embedded Systems*, W. Fischer and N. Homma, Eds. Cham: Springer International Publishing, 2017, pp. 445–467.

[6] S. Parkinson, P. Ward, K. Wilson, and J. Miller, "Cyber threats facing autonomous and connected vehicles: Future challenges," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 11, pp. 2898–2915, Nov 2017.

[7] J. Petit, B. Stottelaar, M. Feiri, and F. Kargl, "Remote attacks on automated vehicles sensors: Experiments on camera and lidar," in *Black Hat Europe*, 11/2015 2015. [Online]. Available: https://www.blackhat.com/docs/eu-15/materials/eu-15-Petit-Self-Driving-And-Connected-Cars-Fooling-Sensors-And-Tracking-Drivers-wp1.pdf

[8] N. Lyamin, A. Vinel, M. Jonsson, and J. Loo, "Real-time detection of denial-of-service attacks in ieee 802.11p vehicular networks," *IEEE Communications Letters*, vol. 18, no. 1, pp. 110–113, January 2014.

[9] J. E. Hachem, T. A. Khalil, V. Chiprianov, A. Babar, and P. Aniorte, "A model driven method to design and analyze secure architectures of systems-of-systems," in *22nd International Conference on Engineering of Complex Computer Systems (ICECCS 2017), Fukuoka, Japan*, 2017, pp. 166–169.

[10] J. E. Hachem, Z. Pang, V. Chiprianov, A. Babar, and P. Aniorte, "Model driven software security architecture of systems-of-systems," in *23rd Asia-Pacific Software Engineering Conference (APSEC)*, Dec 2016, pp. 89–96.

[11] P. Johnson, D. Gorton, R. Lagerstrom, and M. Ekstedt, "Time between vulnerability disclosures: A measure of software product vulnerability," *Computers Security*, vol. 62, pp. 278 – 295, 2016.

[12] C. Guariniello and D. DeLaurentis, "Supporting design via the system operational dependency analysis methodology," vol. 28, no. 1, 2016, pp. 53–69. [Online]. Available: https://doi.org/10.1007/s00163-016-0229-0

[13] L. Wang and X. Liu, "Notsa: Novel obu with three-level security architecture for internet of vehicles," *IEEE Internet of Things Journal*, pp. 1–1, 2018.

[14] V. Katewa, R. Anguluri, A. Ganlath, and F. Pasqualetti, "Secure reference-tracking with resource-constrained uavs," in *2017 IEEE Conference on Control Technology and Applications (CCTA)*, Aug 2017, pp. 1319–1325.

[15] H. Sunghyuck, L. Sunho, and S. Jaeki, "Unified modeling language based analysis of security attacks in wireless sensor networks: A survey," *KSII Transactions on Internet and Information Systems*, 2011.

[16] D. Ding, Z. Wang, G. Wei, and F. E. Alsaadi, "Event-based security control for discrete-time stochastic systems," *IET Control Theory Applications*, 2016.

[17] E. Adi, Z. A. Baig, P. Hingston, and C.-P. Lam, "Distributed denial-of-service attacks against http/2 services," *Cluster Computing*, 2016.

[18] W. Xu, W. Trappe, Y. Zhang, and T. Wood, "The feasibility of launching and detecting jamming attacks in wireless networks," in *6th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2005.

[19] H. Liu and H. Ning, "Zero-knowledge authentication protocol based on alternative mode in rfid systems," *IEEE Sensors Journal*, 2011.

[20] G. N. Khan, J. Yu, and F. Yuan, "Xtea based secure authentication protocol for rfid systems," in *ICCN*, 2011.

[21] L. H. Ning Huansheng and Y. Chen, "Ultralightweight rfid authentication protocol based on random partitions of pseudorandom identifier and pre-shared secret value," *Chinese Journal of Electronics*, 2011.