Kavallieratos *et al*. *J Surveill Secur Saf* 2022;3:128-49
DOI: 10.20517/jsss.2022.17

Journal of Surveillance, Security and Safety

Original Article

Open Access

Risk assessment and control selection for cyber-physical systems: a case study on supply chain tracking systems

Georgios Kavallieratos[1], Christos Grigoriadis[2], Angeliki Katsika[3], Georgios Spathoulas[3], Panayiotis Kotzanikolaou[2], Sokratis Katsikas[1]

[1]Department of Information Security and Communications Technology, Norwegian University of Science and Technology, Gjøvik NO2802, Norway.
[2]Department of Informatics, University of Piraeus, Piraeus GR18534, Greece.
[3]Department of Computer Science and Biomedical Informatics, University of Thessaly, Lamia GR35131, Greece.

Correspondence to: Dr. Georgios Spathoulas, Department of Computer Science and Biomedical Informatics, University of Thessaly, 2-4 Papasiopoulou st., Lamia, GR35131, Greece. E-mail: gspathoulas@uth.gr; ORCID: 0000-0003-2947-486X

How to cite this article: Kavallieratos G, Grigoriadis C, Katsika A, Spathoulas G, Kotzanikolaou P, Katsikas S. Risk assessment and control selection for cyber-physical systems: a case study on supply chain tracking systems. *J Surveill Secur Saf* 2022;3:128-49. http://dx.doi.org/10.20517/jsss.2022.17

Received: 20 May 2022 **First Decision:** 1 Sep 2022 **Revised:** 15 Sep 2022 **Accepted:** 9 Oct 2022 **Published:** 28 Oct 2022

Academic Editor: Richard Brooks **Copy Editor:** Jia-Xin Zhang **Production Editor:** Jia-Xin Zhang

Abstract

Aim: This paper proposes a novel risk assessment methodology for complex cyber-physical systems. The proposed methodology may assist risk assessors in: (1) assessing the risks deriving from cyber and physical interactions among cyber-physical components; and (2) prioritizing the control selection process for mitigating these risks.

Methods: To achieve this, we appropriately combine and modify two recent risk assessment methodologies targeted to cyber-physical systems and interactions as underlying building blocks. By applying the existing methodology, we enable the utilization of well-known software vulnerability taxonomies to extract vulnerability and impact submetrics for all the interactions among the system components. These metrics are then fed to the risk analysis phase to assess the overall cyber-physical risks and prioritize the list of potential mitigation controls.

Results: To validate the applicability and efficiency of the proposed methodology, we applied it in a realistic scenario involving supply chain tracking systems.

Conclusion: Our results show that the proposed methodology can be effectively applied to capture the risks deriving from cyber and physical interactions among system components in realistic application scenarios, while for large-scale networks, further testing should be carried out.

Keywords: Cyber-physical systems, risk assessment, risk mitigation, attack paths, security control selection

1. INTRODUCTION

The digitalization of critical infrastructures enables the dynamic, remote, and efficient monitoring and control of complex cyber-physical systems (CPS) to facilitate their services and operations.

At the same time, it enables new security threats and risks, which may exploit the increased connectivity of CPS, and the underlying vulnerabilities of numerous interconnected system components. As cyber-physical connectivity enables cyber-physical attack paths between interconnected systems, this leads to multiple attack opportunities for attackers. Potential security issues could disrupt critical system functions and services, including critical data and services, or even physical damage to the critical infrastructure. As long as the interconnectivity and interdependency of such infrastructures increases, the propagation of these risks will increase.

The complexity of critical infrastructures and the heterogeneity of CPS employed have introduced several security issues. Supply chain management is a typical instance of heterogeneous CPS with increased security risks. The supply chain infrastructure relies on typical information and communication technology (ICT), as well as operational technology (OT). Additionally, the geographical diversity of systems and the multiple levels of outsourcing increase the risks of cyber attacks. According to NIST[1], the supply chain is defined as the linked set of resources, services, and systems in several levels of an enterprise throughout the product and services life-cycle. Potential security issues in such infrastructure could provoke cascading effects on other systems and infrastructure.

Several security threats, vulnerabilities, and risks have been analyzed for CPS and critical infrastructures[2,3]. Security attacks may propagate within the infrastructure by exploiting the interactions of CPS components, along with their underlying security vulnerabilities, to create complex attack paths. These attack paths are essentially sequences of interacting assets that an attacker can exploit in turn to attack one or more systems within the path[4]. Additionally, several security risks may arise throughout the supply chain in suppliers, their supply chains, and their products or services[1]. Furthermore, the flows and stocks of the supply chain could be the entry points to attack the overall infrastructure[5]. Various security vulnerabilities for supply chain infrastructures have been studied in the literature[1,6]. To address such security issues and improve the security and resilience of such infrastructures, a comprehensive security analysis is needed.

Therefore, the security threats, vulnerabilities, and risks of the supply chain infrastructure need to be analyzed systematically towards a secure and resilient infrastructure. Security risks throughout the supply chain are often undetected, while their impact is significant for the overall infrastructure[1]. Indeed, research on supply chain cyber security has pointed out that several security risks exist and should be adequately mitigated[7,8]. Although several risk management methodologies for supply chain infrastructures exist, the risk propagation within the infrastructure and the optimal control selection to minimize security risks are only partially analyzed[9]. Furthermore, the analysis of the possible attack paths between the infrastructure's components facilitates understanding the cascading effects of a cyber attack.

In this paper, we focus on the risk analysis and risk treatment phases of the risk management process[10] for cyber-physical systems. Particularly, we propose a methodology to analyze security risks considering system vulnerabilities and threats and the aggregate risks between the systems, as well as identifying the optimal security controls to minimize the risks posed by the aforementioned security issues. The proposed methodology leverages the novel risk assessment methodology for assessing IoT-enabled cyber-physical attack paths proposed in[11] and the risk assessment and treatment methodology proposed in[12].

Namely, a comprehensive risk management framework for CPS is proposed combining the steps and phases from the aforementioned approaches. The novel framework includes the steps and phases from both approaches and enables the modeling of interactions between components, the threat and vulnerability analysis, the impact of each threat, and the mitigation of these threats and vulnerabilities by identifying the most appropriate security controls. The contributions of this work are as follows:

- We extend our previous works described in[11,12] towards a comprehensive risk management framework for CPS. Our main goal is to identify and assess the risk derived from the combined cyber and physical interactions, as well as producing the optimal set of security controls to effectively minimize the derived risk.
- We propose a method to analyze the risk propagation in CPS by leveraging the cyber and physical interactions between the system components. By applying and properly modifying the methodology of Stellios *et al.*[11], we enable the utilization of well-known software vulnerability taxonomies, such as CVE and CPE, to extract vulnerability and impact submetrics for all the interactions among the system components.
- We propose a method to systematically select the optimal set of security controls to minimize security risks. The interaction metrics are then fed to the risk analysis engine initially proposed by Kavallieratos *et al.*[13] to assess the overall cyber-physical risks and prioritize the list of potential mitigation controls.
- We apply the proposed methodology in a realistic supply chain tracking CPS to illustrate the workings of the proposed methodology.

The structure of the paper is as follows. Section 2 provides a background analysis of the methods and tools used. Section 3 introduces the proposed methodology. Section 4 illustrates the working of the proposed method in a supply chain use case. Section 5 concludes this work and introduces future research paths.

2. BACKGROUND WORK

Various risk assessment and risk management frameworks can be found in both gray and scientific literature, which are also applicable to cyber-physical systems. First, we briefly describe threat, vulnerability, and security control models, frameworks, and scoring systems that have been proposed by standardization bodies and/or other relevant organizations. Although these works belong to the gray literature, these standards are widely accepted and are often used as de facto standards. Then, we briefly describe risk assessment methodologies recently proposed in the literature for cyber-physical systems. Finally, we focus on two CPS risk assessment methodologies, which serve as the underlying building blocks for the proposed methodology.

2.1. Basic building blocks

We briefly describe some well-known security models, including vulnerability catalogs and threat and risk assessment models, which are utilized as building blocks in selected phases of the proposed methodology. In Section 3, where the proposed methodology is presented, we explain how these blocks are utilized in specific phases.

2.1.1. CPE

The Common Platform Enumeration[14] catalog is a structured naming scheme for information technology systems, software, and packages. Utilizing CPE-recorded products makes it easier to map the various components residing in systems, and various scanners used for enumeration utilize the CPE catalog for their output.

Finally, for each product recorded in the CPE catalog, linked entries might reside in the CVE and CWE catalogs. In the proposed methodology, we utilize CPE to assist in the mapping of CPS components to actual vulnerabilities.

2.1.2. CVE

The Common Vulnerabilities and Exposures (CVE)[15] list contains publicly known cybersecurity vulnerabilities. A vulnerability is defined as a "weakness in the computational logic (e.g., code) found in software and hardware components that, when exploited, results in a negative impact to confidentiality, integrity, or availability". Each CVE entry, i.e. vulnerability, contains an identification number, a description, and at least one reference for publicly known cybersecurity vulnerabilities. Additional entry information can include mitigation information, severity scores, and impact ratings according to the Common Vulnerability Scoring System (CVSS). We utilize CVE in the identification of low-level software vulnerabilities for the system components.

2.1.3. CVSS

The Common Vulnerability Scoring System (CVSS)[16] provides a framework and a methodology, through which security experts and vulnerability researchers may define specific *exploitability metrics* of a vulnerability that include: The attack vector (AV) that is required to exploit the vulnerability, with possible values physical (P), local (L), adjacent network (A), and network (N); the attack complexity (AC) required for exploiting the vulnerability, either low (L) or high (H); the privileges required (PR) to exploit the vulnerability, taking the value none (N), low (L), or high (H); the required user interaction (UI) with possible values none (N) and required (R); and finally the scope (S), which denotes whether the vulnerability affects only the system in question or changes to affect other software. Furthermore, the *impact metrics* define the potential effect on confidentiality (C), integrity (I), and availability (A), all taking the value none (N), low (L), or high (H). The available labels for these characteristics are supported by weights, which can be utilized to produce numerical scores reflecting the severity and impact of the vulnerability. These scores can then be translated into a qualitative representation to help organizations properly assess and prioritize their vulnerability management processes. In our methodology, we extensively use CVSS to measure and map the cumulative vulnerability level for each identified interaction between system components.

2.1.4. DREAD and STRIDE

DREAD[17] is a cybersecurity risk assessment approach that stands for *damage*, *reproducibility*, *exploitability*, *affected users/systems*, and *discoverability*. The aforementioned aspects characterize the risk associated with different attack scenarios, and in particular, they aim to answer questions such as "what is the damage of a potential attack or threat to a system", "how an attack may be reproduced", "how easy an adversary may attack to a system", "how many systems or people may be affected", and "how easy it is for the attacker to identify the relevant vulnerabilities". Additionally, STRIDE[18] is a threat modeling approach that facilitates the process of threat identification and analysis of six types of threats: *spoofing* (violation of authenticity), *tampering* (violation of integrity), *repudiation* (violation of non-repudiability), *information disclosure* (violation of availability), and *elevation of privileges* (violation of authorization). Both approaches have been developed by Microsoft aiming at the comprehensive security analysis of the targeted systems, since STRIDE identifies and analyzes the cybersecurity threats, while DREAD analyzes the accordant risks of the aforementioned cybersecurity threats. Although both approaches have been developed to analyze the security of software-related systems, their application in domains where the application of CPS is prominent is popular[19–21]. In the proposed methodology, we utilize STRIDE in threat analysis and DREAD in risk evaluation and propagation.

2.1.5. Genetic algorithms

Genetic algorithms (GAs) facilitate the randomized search by leveraging the structures of natural genetics and natural selection mechanisms[22]. Particularly, GAs aim to solve problems for which the solution cannot be found with exhaustive search mechanisms due to the very large solution space. The main attributes of GAs are the coding scheme, a set of operators, and a fitness function[23]. Overall, a GA addresses unconstrained optimization problems, including the one this study aims to address.

2.2. Related Work on CPS risk assessment methodologies

Various methods have been proposed for CPS risk assessment. Kott *et al.*[24] proposed a method that uses mission impact assessments as a tool to assist operational decision makers in applying cyber defense security controls. They defined a simulation platform that can emulate CPS, such as water and energy distribution systems. The analysis demonstrates that, based on the simulation platform, it is possible to discover various hidden dependencies which require risk treatment. Lyu *et al.*[25] proposed a cyber-to-physical (C2P) risk assessment model based on hierarchical Bayesian networks (BN). The model estimates the probability of a security incident compromising a target based on CVSS metrics to produce quantitative risk values, which are also assessed through qualitative risk analysis. Tantawy *et al.*[26] presented an integrated approach for the analysis and design of a cyber security system for a given CPS, where the physical threats are identified first to guide the risk assessment process.

Some works are targeted to specific CPS application environments. For example, Abie and Balasingham[27] proposed an adaptive risk-based security framework for CPS for the health sector. The goal is to assess the potential damages and future benefits using game theory and machine learning techniques, which, in turn, enables the security mechanisms to adjust their security decisions accordingly. Seale *et al.*[28] presented a risk assessment framework for medical infrastructures. Their framework uses various building blocks that we also apply, such as STRIDE, CVE and CVSS. Furthermore, Mokalled *et al.*[29] proposed a general risk management framework for CPSs focusing on specific characteristics of the CPSs. Rosado *et al.*[30] proposed a risk analysis approach for CPS based on MARISMA. However, this approach focuses on general elements of the CPS. Additionally, defensive strategies and policies in CPS towards increasing the level of defense for the communication network in a ship infrastructure were proposed by R. Sahay *et al.*[31]. The propagation of the attacks between CPSs is studied in[32] considering the sensitivity of the assets and the impact of the attacks.

Security risks in supply chain infrastructure have been extensively analyzed in the literature[33,34]. Ho *et al.*[9] conducted a literature review of existing methodologies to manage risks in supply chain infrastructures. Supply chain systems are cyber-physical systems where both safety and cyber security risks may arise[35]. To this end, a comprehensive risk assessment methodology is needed to analyze both safety and security risks and the accordant controls. Furthermore, the analysis of the risk propagation between the supply chain systems facilitates the identification of optimal sets of security controls.

The proposed methodology combines and extends two recent cyber-physical risk assessment methodologies. We focus on those underlying methodologies, and below we will briefly describe them and their advantages and disadvantages.

2.2.1. A methodology to assess IoT-enabled, cyber-physical attack paths[11]

The methodology of Stellios *et al.*[11] is a CPS-specific risk assessment methodology, whose goal is to assess cyber-physical attack paths which are IoT-enabled, i.e. the attack involves the exploitation of inter-connected IoT-IT systems. The methodology is *target-oriented* and *source-driven*. For each run of the methodology, a critical system is defined as the attack target. For all the systems/devices in range, their interactions, i.e. connectivity or other dependencies, are defined. Then, based on the interactions, a recursive algorithm is executed to construct all the attack paths and thus identify all the effective sources for all the attack paths. To assess the vulnerability of each interaction and each attack path, it utilizes CVSS-like vectors to indentify specific interactions and assess their exploitability characteristics in cyber-physical systems. Ultimately, the vulnerability level of components is also calculated in a CVSS vector and compared with the interaction capabilities to produce validated results. Furthermroe, this approach enables the assessor to map and express all the interactions in a

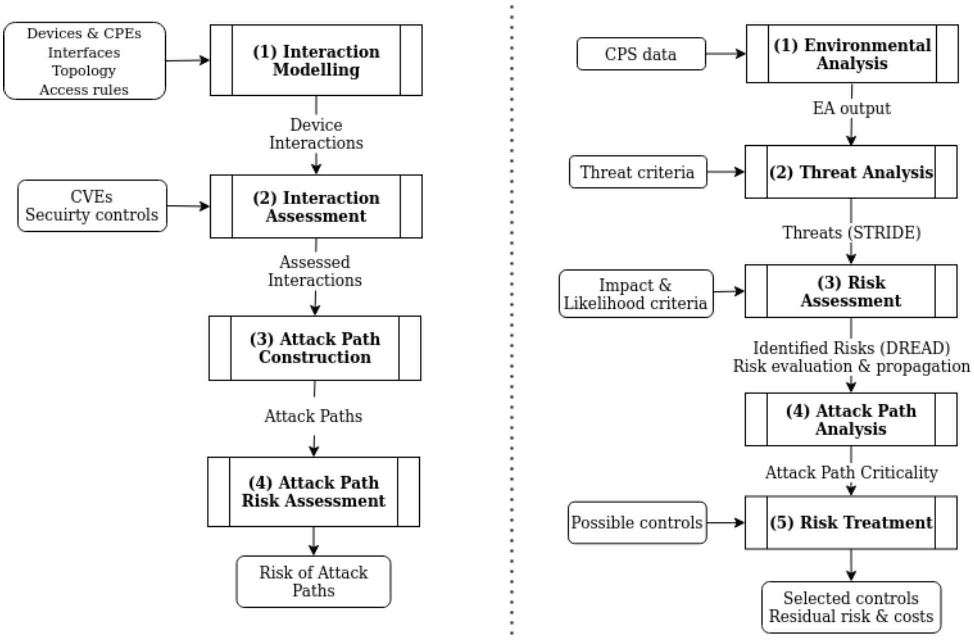

Figure 1. An overview of the underlying CPS RA methodologies of Stellios *et al.* [11] (left) and Kavallieratos *et al.* [13] (right).

source–destination format„ which are then combined to generate probable attack paths to later be assessed.

As illustrated in Figure 1, the methodology consists of four phases. In the first phase, the interaction modeling is executed, which aims to map all of the potential cyber and physical interactions that reside in the context of the infrastructure under duress and target towards a designated system. The algorithm developed for this phase essentially iterates over cataloged device and network classes, which contain connectivity configurations and input/output capabilities, to enumerate applicable interactions amongst the components. In the second phase, the calculated interactions are further assessed based on the vulnerabilities of the target component. Essentially, the previously enumerated interactions are extended with a so-called *cumulative vulnerability vector* (CVV), derived from a combination of the CVEs identified for the active devices acting as targets. This cumulative score is refined by environmental information such as networking information and related security controls. In the third phase, the attack paths are built through the exhaustive combination of the assessed interactions enumerated in Phase 2. This algorithm builds attack paths that may vary in length, involving one or more interactions; in this case, loops that do not offer elevated exploitability characteristics are considered as noise and removed. In the fourth phase, the previously enumerated attack paths are assessed and a risk value per path is assigned. This risk is produced using the CVV of each interaction tuple, the vulnerabilities of the initial nodes, and the characteristics of adversaries that are applicable to the infrastructure and capable of initiating the attack path.

On the positive side, the methodology in [11] allows for detailed cyber and physical interaction modeling. In addition, it takes into account low-level vulnerability input by utilizing de facto standards such as CPE, CVE, and CVSS. Finally, by targeting the analysis to a specific "critical target system", it allows for an efficient computation of risks for all the attack paths that may lead to the critical target system. On the negative aspects, it does not support detailed risk management and control selection, which is only possible through a repeated "what-if" analysis that requires sequential re-execution of the methodology.

2.2.2. A CPS methodology for risk propagation and control selection[13]

Kavallieratos *et al.*[13] proposed a methodology to analyze the propagation of the security risks among complex CPS infrastructures. By leveraging a genetic algorithm approach, the most effective and efficient security controls per CPS component are identified. Building upon the results of the aforementioned work, we extend the methodology to analyze attack paths between CPSs and improve the security control selection process to identify a set of security controls that minimizes both the residual risk and the cost[12]. Overall, the phases of the aforementioned works are the stepping stones for the methodology proposed herein.

In the environment analysis, the system model is described in terms of components. The CPS components are identified along with their interconnections, dependencies, and inter-dependencies. By leveraging this information, the CPS coefficients are estimated and their values represent the effects of each component on the other components and the overall system. The rest of the process defined as Cybersecurity analysis consists of different stages of the risk management process, as specified in ISO 31000[10]. In the Threat analysis stage, the cybersecurity threats are identified per component and CPS using the STRIDE methodology. In the Risk assessment stage, the risk value associated with each threat from the previous step is estimated by leveraging the DREAD methodology. Having estimated the risk values, the risk per component, CPS, and path is evaluated. Consequently, in the Attack path analysis step, the risk aggregation between components and CPSs are analyzed, and the most critical attack paths are identified. Finally, in the Risk treatment step, a genetic algorithm s decides the most effective and efficient set of cybersecurity controls. The *applicability*, *effectiveness*, and *cost* of each control is considered.

The aforementioned approach analyzes the target environment following a system of systems perspective. Namely, the security analysis of each component is performed towards the overall analysis of the targeted ecosystem, also considering the total effect of each component on the targeted ecosystem. This approach facilitates the analysis of ecosystems under development to ensure their overall security. However, the correlations between the components are estimated considering only the information and control flows without taking into account cyber interactions. Additionally, the system and component vulnerabilities are not considered in this approach.

3. THE PROPOSED METHODOLOGY

We combine and properly modify the above-mentioned methodologies[11,13] to allow for dynamic and efficient risk analysis and risk treatment methodology for complex CPS.

The proposed methodology utilizes in the various phases and properly combines, as building blocks, the security models described in Section 2.1. By utilizing the interaction modeling (see Phase 1) and interaction assessment (see Phase 2) phases of Stellios *et al.*[11], we model both the cyber and physical interactions between the components in a detailed manner.

In phase 1, we utilize existing asset catalogs to map assets in the form of CPE entries. In Phase 2, the CVE catalog is utilized to map vulnerabilities to assets, while the CVSS scoring system is extensively used in the interaction assessment phase.

The threat analysis (see Phase 3) is based on the relevant component of Kavallieratos *et al.*[13] and follows the STRIDE threat model. The output of this phase is mapped to the DREAD risk assessment model to assist later in the control selection process.

By combining the threat input with the vulnerability and impact interaction assessment metrics, system risks are computed for the whole system under study by extending the relevant phase in[13] (Phase 4). The risk

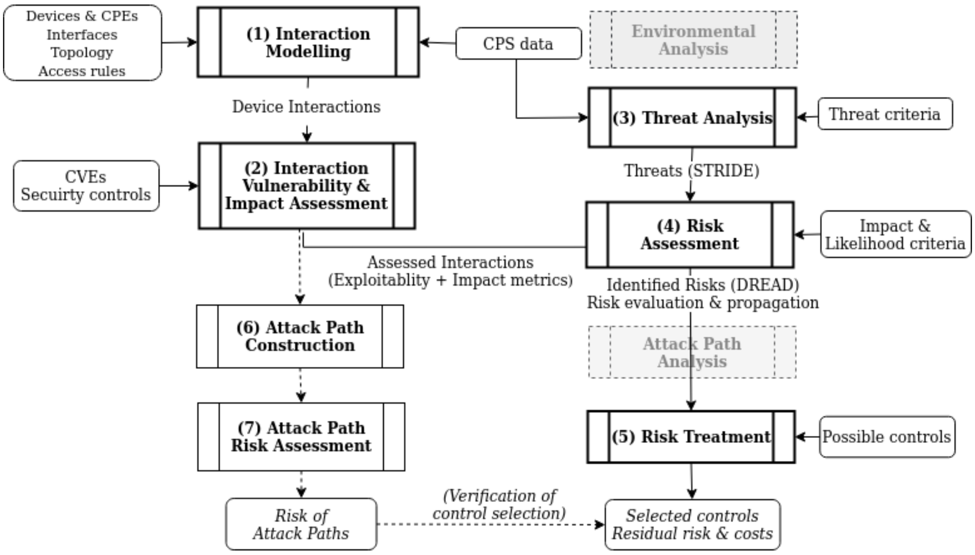

Figure 2. The proposed CPS methodology extends those in [11,13] to take into account detailed vulnerability information and existing interactions while also assessing mitigation controls.

treatment (Phase 5) is again based on the relevant phase in [13], where the effect of detailed security control selection can be evaluated. Risk treatment utilizes a genetic algorithm approach to allow the prioritization of possible security controls.

Finally, as a verification of the control selection, the attack path construction (Phase 6) and attack path assessment (Phase 7) in [11] may be optionally computed for chosen target components to verify the effectiveness of the control selection process. In the following, we analyze the proposed methodology in detail. The proposed methodology is depicted in Figure 2.

To assist the reader in mapping the relevant phases and flow of the proposed methodology with respect to the underlying methodologies [11,13], we depict in bold frames the main phases of the methodology (Phases 1-5) and in light frames the optional phases (Phases 6 and 7). Finally, for easier reference with the underlying methodologies shown in Figure 1, those phases that are not used in the proposed methodology are grayed out.

3.1. Interaction modeling

Following our previous approach [11], we model both the cyber and physical interactions by mapping their physical channels, networks, and logical access. We denote an interaction between a source node (asset) x and a destination node y as $Int(x, y, type)$, which represents the effect that x may have on y based on their proximity and connectivity. We define two categories of interactions, *physical* and *cyber*, where each category defines specific *interaction types*.

Cyber interactions include all the actions that may be triggered by the source towards the destination node due to their cyber connectivity. Nodes that have cyber interactions belong to either the same local network (Adjacent network in the CVSS terminology) or different networks. In addition, the source node x may have various access rights in the destination of the interaction y, ranging from no access (i.e., x and y are simply nodes in the same network), low-level access (e.g., x may have user-level access on y), or full access (i.e., admin rights). Table 1 summarizes the cyber interaction types that we adopt from [11].

Then, utilizing the proximity of systems and their interface types as input enables us to enumerate their *physical interactions* by iterating over an exhaustive combination of those elements. The physical attack vector (AV:P)

Table 1. Cyber interaction types C1-C6, based on the connectivity and the logical access of x to y

Connectivity	Logical access None (no explicit access)	Low (user-level)	High (admin-level)
L2 (local) network	**C1**	**C2**	**C3**
L3 (remote) network	**C4**	**C5**	**C6**

Table 2. Physical interactions types defined based on the physical proximity (P1), wireless I/O proximity (P2), and network proximity (P3) of shared-band network interfaces

Type	Description	Interface	Examples
P1	**Physical proximity** (x may use a moving part and/or moving capabilities to physically reach y)	Remotely controlled moving parts or devices	Robotic arm, crane, wheeled device, drone
P2	**Wireless I/O proximity** (x is in range with a wireless I/O interface of y)	Audio, visual, optical interfaces	Line-of-sight (LiDAR, IR), audio/video interfaces
P3	**Networks' proximity** (x and y at *different* networks that are in range)	Different, but shared-band wireless interfaces	802.11.x and 802.15.x operate at 2.4 GHz

described in CVSS is applied for machine-to-machine interactions that are able to reach each other physically. In addition, AV:A is considered appropriate for physical interaction types P2 and P3, since adjacent network access is adequate for physical interactions that require network proximity. We define three types of physical interactions, as shown in Table 2.

3.2. Interaction vulnerability and impact assessment

To analyze the vulnerability and impact level for each interaction, we properly modify the interaction assessment phase of our previous work[11]. We briefly refer to the underlying methodology and the work in[11] for a detailed analysis.

For each identified cyber and physical interaction, a CVSS vector is assigned based on predefined values (see Table 3). The CVSS-like vector presented in Table 3, called $IntCVSS$, corresponds to the *implied capabilities* that the source node x has on the destination node y, due to their interaction type[1]. We provide some examples to clarify how each $IntCVSS$ vector is derived (see[11] for the complete process). Furthermore, in Section 2.1, we provide a brief summary of the CVSS metrics and values.

For example, a cyber interaction of type C3 implies that the source and the destination nodes reside in the same network and the source node x has admin-level access to the destination node y (see Table 1). This is reflected in the $IntCVSS$ vector through the metrics AV and PR and the C/I/A Impact, which are, respectively, adjacent network (AV:A), high implied privileges (PR:H) and high impact for all types (C:H/I:H/A:H). Similarly, physical interaction of type P3 implies that, while the source and the destination nodes are connected in different networks, they are equipped with shared band frequency interfaces and are within range [Table 2]. Therefore, x has the implied capability of performing jamming or injection attacks on y. This is reflected by the adjacent network attack vector (AV:A), no implied access privileges (PR:N), and the low integrity and availability impact (I:L/A:L).

As explained above, the $IntCVSS$ vector denotes the initial capabilities that the source node x has on the destination node y due to their interaction. However, this "inherited" relation may be extended by the source node by exploiting the underlying vulnerabilities that exist on node y. As mentioned in Section 2, the CPE identifiers of all the nodes (assets) are utilized to retrieve all the underlying vulnerabilities of each node. Each vulnerability found is examined as an isolated (single) vulnerability. In addition, all the vulnerabilities that correspond to each node are combined for each different asset (CPE id) using the vulnerability chaining process

[1] The cyber and physical interaction types are illustrated in Tables 1 and 2, respectively

Table 3. $IntCVSS$**: predefined implied capabilities for all interaction types in the form of CVSS vectors. To interpret the values, refer to section 2.1**

	Exploitability metrics					Impact metrics		
Interaction type	**AV**	**AC**	**PR**	**UI**	**S**	**C**	**I**	**A**
C1	A	H	N	N	U	N	N	N
C2	A	H	L	N	U	L	L	L
C3	A	H	H	N	U	H	H	H
C4	N	H	N	N	U	N	N	N
C5	N	H	L	N	U	L	L	L
C6	N	H	H	N	U	H	H	H
P1	P	H	N	N	U	N	L	L
P2	A	H	N	N	U	L	L	L
P3	A	H	N	N	U	N	L	L

Table 4. Summary of all vectors utilized in interaction assessment

$IntCVSS(x, y, type)$	A CVSS-like capability vector assigned on the interaction based on the interaction's type, by applying section 2.1.. The $IntCVSS$ may also be transformed based on the environment by considering relevant network security controls if such information is available.
$\{SingleCVSS_y\}$	A list of all the single CVSS vectors corresponding to vulnerabilities identified in y.
$\{ChainedCVSS_y\}$	A list of all the CVSS vectors of the chained vulnerabilities of y.
$CVV((x, y, \text{type}))$	The resulting Cumulative Vulnerability Vector of the interaction as defined in Equation (1).

Table 5. STRIDE per system

System per threat	**S**	**T**	**R**	**I**	**D**	**E**
System 1			x		x	
System 2	x			x	x	

described in the CVSS standard[36] to produce the chained vulnerability vector of each node.

Based on the above information, the cumulative vulnerability vector (CVV) of an interaction $CVV((x, y, \text{type}))$ is defined as a unique CVSS-like vector, representing the implied capabilities of the source node and the existing vulnerabilities of each interaction. Table 4 summarizes all the input vectors used for the computation of the CVV, while Equation (1) denotes the computation formula. The output vector can then be easily transformed to a numerical value in the range of 0–10, as defined in the CVSS standard.

$$CVV((x, y, \text{type})) = V \in (IntCVSS(x, y), SingleCVSS_y, ChainedCVSS_y) \;\; \texttt{s.t.:} \quad \begin{cases} \mathrm{C} \geq \mathrm{L} \,\&\, \mathrm{I} \geq \mathrm{L} \,\&\, \mathrm{A} \geq \mathrm{L} \\ \texttt{V has max(Impact,Exploitability)} \end{cases} \tag{1}$$

3.3. Threat analysis

The analysis of the risk propagation requires the identification of potential security threats that the risks may arise. To this end, the STRIDE methodology is utilized in the phase of the proposed methodology. By leveraging the six threats (**s**poofing, **t**ampering, **r**epudiation, **i**nformation disclosure, **d**enial of service, and **e**levation of privileges), the attack scenarios of each component/system of the targeted environment are analyzed. STRIDE provides a qualitative threat analysis, and the results are used as input to the DREAD methodology for a comprehensive risk analysis. Furthermore, this technique facilitates the identification of threats per element (system), as shown in Table 5.

3.4. Risk analysis

In this phase, the quantitative risk analysis is performed, considering the results from the previous phases. Particularly, the DREAD methodology is utilized aiming to quantify specific aspects (**d**amage potential, **r**eproducibility, **e**xploitability, **a**ffected systems, and **d**iscoverability) of STRIDE threats and attacks to assign meaningful numbers to the elements of risk by means of Equations 2-4.

The risk value R associated with each STRIDE threat $t \in \{S, T, R, I, D, E\}$ for system s is calculated by using the following formulas[17,37]:

$$Impact_t^s = \frac{Damage + Affectedsystems}{2}, \tag{2}$$

$$Likelihood_t^s = \frac{Reproducibility + Exploitability + Discoverability}{3}, \tag{3}$$

$$Risk_t^s = \frac{(Impact_t^s + Likelihood_t^s)}{2}. \tag{4}$$

$Impact_t^s$ describes the effect of a cyber attack realizing specific threat t upon a component s, while $Likelihood_t^s$ describes the probability of the specific threat t being realizing in s.

3.4.1. Risk propagation

The *aggregate* risk $R_t^{agg_{c_j}}$ of component c_j is calculated using Equation (5).

$$R_t^{agg_{c_j}} = \max(R_t^{dir_{c_j}}, R_t^{prop_{c_j}}), \tag{5}$$

where *direct* risk $R_t^{dir_{c_j}}$ is the risk of c_j without considering the possible connections with other components and is estimated using Equations 2-4, while the *propagated* risk $R_t^{prop_{c_j}}$ is calculated considering the connections to other components that c_j has. The fraction of the impact that an event has on any c_k on any path p_l from c_i to c_j is represented by $eff_{p_l}^T$ and is calculated as

$$eff_{p_l}^T = \prod_{i=1}^{j-1} eff_{c_i c_{i+1}}^T. \tag{6}$$

where $eff_{c_i c_{i+1}}^T$ values are extracted from the CVV interaction scores of Equation (1) after the values have been normalized in the range [0,1].

The risk propagated over path p_l, originating at component c_i and terminating at component c_j, is calculated by:

$$R_t^{prop_{c_j}^{p_l}} = \frac{eff_{c_i c_j}^{T_{p_l}} * Impact_t^{c_i} + L_t^{c_i}}{2}. \tag{7}$$

The whole system is described by c_0 and the *global* risk of threat t for the system is calculated by:

$$R_t^s = R_t^{agg_{c_0}} = \max(R_t^{dir_{c_0}}, R_t^{prop_{c_0}}), \tag{8}$$

where the direct risk for the system is not applicable ($R_t^{dir_{c_0}} = 0$) and the propagated risk for the system is calculated as for any other node ($R_t^{prop_{c_0}} = \max_{p_l} R_t^{prop^{p_l}_{c_0}}$). Thus,

$$R_t^s = \max_{p_l} R_t^{prop^{p_l}_{c_0}} \tag{9}$$

To summarize, our risk analysis method is able to capture the effect of both the cyber and physical interactions between components. This is due to the fact that the CVV value of each interaction has the ability to chain vulnerabilities and interactions of different types, as illustrated in Equation (1). The CVV is then utilized in Equation (6) to compute the coefficient $eff^T_{c_ic_{i+1}}$. The paths along which the risk is propagated consist of potential one-to-one interactions between different system components, which can be physical or cyber-related. As the global risk computed in Equations 8 and 9 is based on the propagated risk over paths [Equation (7)], this enables our model to capture the impact reflected from the cyber to physical components and vice versa.

Further details about the method used and the aforementioned equations are omitted in the interest of saving space and can be found in[12,13].

3.5. Risk treatment

The purpose of the risk treatment phase is twofold. First, the identified risk is to be minimized. Second, the propagation of the risk within the infrastructure is to be mitigated. Particularly, the proposed approach leverages a set of security controls that are appropriate for the targeted system. The effectiveness and the cost of each control are considered in selecting the most appropriate security control. The effectiveness of the controls reflects the effect of each control for each threat and each component c_i. The values of $Impact_t^{c_i}$ and $Likelihood_t^{c_i}$ are affected by the effectiveness of controls and, therefore, the risk per system and to the overall infrastructure. Furthermore, the cost $Cost_m$ of each control m is considered. For a system with N components and a list with M controls with the cost vector $C = [cost_1, cost_2, ..., cost_M]$, the following binary matrix AC compactly depicts the applied controls throughout the system:

$$AC = \begin{bmatrix} ac_{1,1} & ac_{1,2} & \dots & ac_{1,N} \\ ac_{2,1} & ac_{2,2} & \dots & ac_{2,N} \\ \dots & \dots & \dots & \dots \\ ac_{M,1} & ac_{M,2} & \dots & ac_{M,N} \end{bmatrix}, \tag{10}$$

where

$$ac_{i,j} = \begin{cases} 0, \textit{if control i is not applied to component j} \\ 1, \textit{if control i is applied to component j} \end{cases}. \tag{11}$$

The total cost TC_{AC} of the applied controls solution AC is $TC_{AC} = AC * C$.

The approach proposed in[13] aims to provide separate lists of security controls taking into consideration STRIDE threats. The proposed methodology analyzes all the controls identified for all systems, to produce an effective set of security control for the overall infrastructure. The process that enables the elicitation of controls that are effective for more than threats is proposed in[12]. Namely, the process takes for granted that the controls identified in the previous steps are implemented for each threat by leveraging a cascading application of the genetic algorithm approach. For instance, the application of a single control to a specific component could reduce the overall system risk for more than one threat. In terms of global implementation cost, the chocie is more efficient. The proposed scheme supports the identification of the set of controls that minimizes the risk all threats and the implementation cost of the system, as depicted in Figure 3.

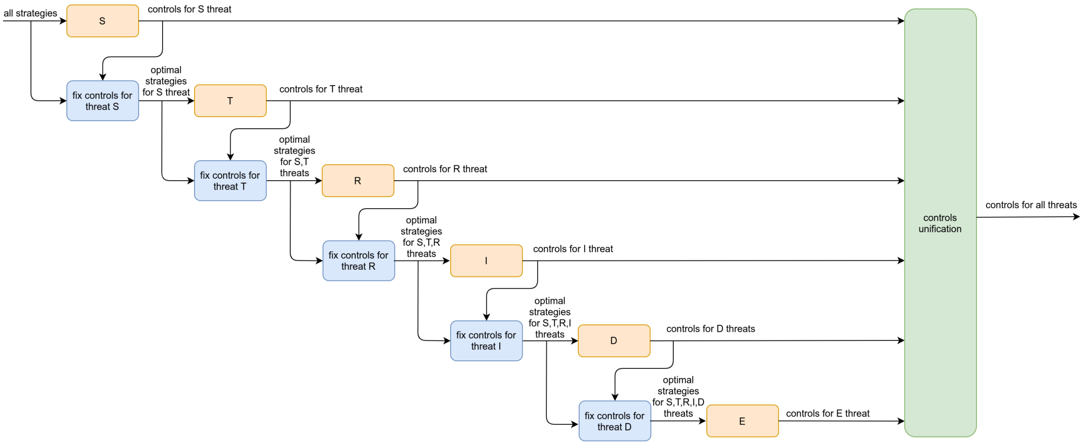
all strategies
S
controls for S threat
optimal strategies for S threat
fix controls for threat S
T
controls for T threat
optimal strategies for S,T threats
fix controls for threat T
R
controls for R threat
optimal strategies for S,T,R threats
fix controls for threat R
I
controls for I threat
optimal strategies for S,T,R,I threats
fix controls for threat I
D
controls for D threats
optimal strategies for S,T,R,I,D threats
fix controls for threat D
E
controls for E threat
controls unification
controls for all threats

Figure 3. Cascading GA process [12].

The concept upon which the approach is based is analyzed in detail in [12]. After applying the genetic algorithm for each threat, the resulting controls are fixed in the set of available controls used as input for the rest of the threats. After all threats have been analyzed, the resulting controls are unified into the optimal set of cyber security controls for the entire system.

Specifically, for each threat, an instance of a genetic algorithm is run to decide the optimal control selection for that threat. This amounts to selecting the set of controls AC that minimizes the system *residual* risk $R^s_{t_{AC}}$ at the lowest total cost TC. For each threat, the controls selected for the threats analyzed previously are fixed as enabled in the AC array. The large size of the search space (all candidate solutions) prohibits the use of exhaustive search methods. Hence, a heuristic optimization method has to be employed [38]; we select a genetic algorithm, even though any other heuristic optimization method would apply in principle.

The design parameters of the genetic algorithm are as follows:

- The search space comprises all possible combinations of controls applied to components.
- Each individual solution is represented by the matrix AC, which is transformed into a binary vector of size $M * N$. The value of each element of the vector represents the decision to apply a specific control to a specific component or not.
- The fitness function is defined as $fit(AC) = R^s_{t_{AC}} + C_{norm}(AC)$, where $C_{norm}(AC) = \frac{TC_{AC}}{TC_{max}}$, with TC_{max} being the largest possible cost of applying all available controls to all system components.
- The initial population size is 100.
- The mutation probability is 0.1.
- The next generation is determined by uniform crossover, with crossover probability equal to 0.5, an elite ratio of 0.01, and 0.3 of the population consisting of the fittest members of the previous generation (i.e., the parents).
- The algorithm terminates when the maximum number of allowed iterations is used. This number is calculated as $iter_{max} = 50 * \sum_{i=1,j=1}^{i=M,j=N} ac_{ij}$.

3.6. Attack path construction and attack path risk assessment

The last two phases of the proposed methodology may be optionally executed to validate the effect of the control selection produced by the Risk Treatment phase. The Attack Path Construction and Attack Path Risk Assessment phases are based on the relevant phases defined in the methodology of Stellios *et al.* [11]. Our goal here, however, is not to prioritize the attack paths but to validate the effect of the controls selected in the Risk Treatment phase in effectively reducing the risk of the critical attack paths. To construct the attack paths, the

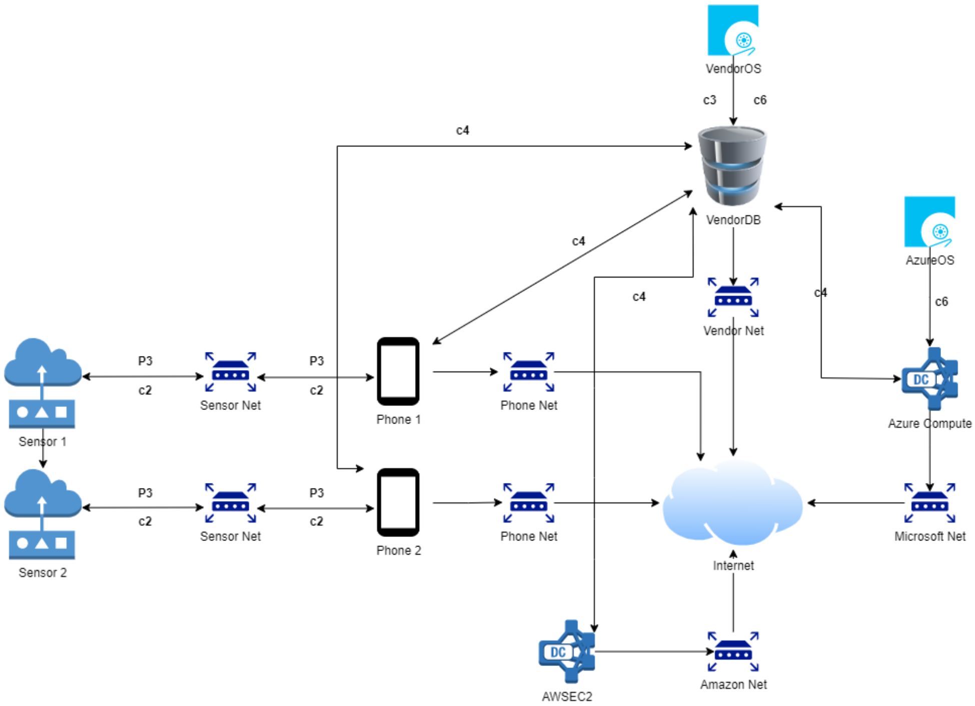

Figure 4. Supply chain scenario.

assessor must first select which asset is set as the critical target system. All the attack paths towards the critical target system are then computed and their risk is calculated as in[11]. Then, the security controls derived from the Risk Treatment phase can be considered in "what-if" scenarios, and the attack path assessment is re-run to verify if the selected security controls have effectively dealt with the highest risk paths towards the critical targets. The selection of a different target system will require re-computing the relevant attack paths and their risk, as well as the risk of the attack paths after the security controls have been considered in the input (i.e., vulnerabilities and interaction types).

4. METHODOLOGY APPLICATION

4.1. Application scenario: supply chain tracking system

To validate the proposed methodology, we applied it in a typical CPS system from supply chain management, mainly supply chain tracking systems. The scenario and the environment mapping were built and adjusted based on a real implementation and the input provided by different stakeholders involved in a common supply chain, such as producers of transferred goods and tracking service providers. We established a typical asset and network map per organization, containing any observed assets or networks that contribute to the supply chain tracking service, and based on the provided input, we implemented the interaction modeling phase of the proposed CPS methodology presented in Section 3. The system under examination contains typical CPS assets used to support logistic functionalities [Figure 4]. The sensors are installed in moving trucks with the purpose of sensing data related to the state of the transferred products, such as current temperature and humidity. The data are then forwarded to an edge device, typically a smartphone acting as the sensors' gateway. The accumulated data are passed from the gateway to a local database hosted by the logistics vendor and two cloud computing services, Amazon Elastic Compute Cloud (Amazon EC2), which offers a broad computing platform, and Azure Compute, which offers a massive range of infrastructure as a service (IaaS) facilities depending on the computation needs of the functionality.

Table 6. CPE identifiers of assets and enumerated vulnerabilities

Asset name	CPE	Relevant CVEs
Sensor	cpe:2.3:h:teltonika-networks:trb245:-:*:*:*:*:*:*:*	CVE-2020-5771, CVE-2020-5772, CVE-2020-5785 , CVE-2020-5787
Phone	cpe:2.3:h:xiaomi:mi_5s:-:*:*:*:*:*:*:*	CVE-2018-20823
AWS compute	cpe:2.3:o:debian:debian_linux:10.0:*:*:*:*:*:*:*	CVE-2022-21682, CVE-2022-26846, CVE-2022-26847, CVE-2022-24301
Vendor DB	cpe:2.3:a:oracle:mysql:5.7.0	CVE-2021-2226, CVE-2021-2202, CVE-2022-21304, CVE-2021-2171
Vendor OS	cpe:2.3:o:debian:debian_linux:10.0:*:*:*:*:*:*:*	CVE-2022-21682, CVE-2022-26846, CVE-2022-26847, CVE-2022-24301
Azure OS	cpe:2.3:o:microsoft:azure:-:*:*:*:*:*:*:*	CVE-2019-0816
Azure compute	cpe:2.3:a:microsoft:azure_stack:-:*:*:*:*:*:*:*	CVE-2019-1234

To calculate the applicable interactions among the various assets within the presented scenario, the algorithm that supports Phase 1 was iterated over the recorded instances of available devices and executed a set of conditions that consider network connectivity, device proximity, and access level from asset to asset. Finally, the identified assets were correlated with the corresponding identifiers from the CPE catalog, which were then utilized to index the corresponding vulnerabilities in the form of CVEs retrieved from the National Vulnerability Database.

4.2. System security analysis (supply chain)

4.2.1. System threat and vulnerability analysis (supply chain)

To initiate and execute the system threat and vulnerability analysis, a combination of the calculated interactions and the enumerated vulnerabilities was implemented. For the destination nodes of the recorded interactions, the applicable vulnerabilities were initially enumerated in a list as SingleCVSS vectors, and then the same vulnerabilities were combined to produce ChainedCVSS vectors, which were also added to the list. From each list, a single or chained vulnerability with the highest exploitability and impact values was selected to act as the CVV. Table 6 illustrates the CPE identifiers of the assets included in the scenario along with a set of enumerated vulnerabilities that acted as the SingleCVSS vectors.

Table 7 contains the initial interaction vector $IntCVSS$ (modified based on existing network security controls), the resulting CVV of the interaction resulting from $IntCVSS$, and the existing vulnerabilities, as described by Equation (1). For each CVV value, we present its CVSS vector, the exploitability and impact subscores, and the overall score. In the context of this table, we observe a set of interesting interactions: For the interaction (Sensor, Phone, C2), we observe that the sensor has write privileges in the database of the phone application that forwards the data to the VendorDB. If the sensor is compromised, this data flow can be abused to inject malicious messages into the phone. For interaction (Phone, Sensor, P3), we observe that both the sensor and the phone utilize Bluetooth frequencies; in this spectrum, both jamming and packet injection attacks are applicable. By compromising the phone in this case, an attacker could both send falsified data to the VendorDB about the sensor readings and send confirmations to the sensor that the data are received. Furthermore, by installing a small jammer on the truck, an attacker could disrupt the communication throughout transfer, which could result in product spoilage. This fact is confirmed by the high values of the calculated integrity and availability impact for the observed interaction. For interaction (AzureOS,AzureCompute,C3), we observe that all the initial impact values are mitigated from high to low based on the existing network security controls; nevertheless, the impact is increased due to the vulnerabilities found on the destination system and the final CVV impact is shifted to high. A similar case holds for the interaction (VendorDB,AzureCompute,C4). Finally, for interaction (VendorDB, Phone, C4), we observe that an attacker that has compromised the phone network, through either the phone or the service provider, could intercept and inject the confirmation messages sent from the vendor database towards the phone application. In such a scenario, neither the phone application nor the VendorDB would be able to identify the loss of integrity of the relevant supply chain tracking data.

We observe that, while interactions (VendorDB,AzureCompute,C4) and (VendorDB, Phone, C4) share the same interaction type and initial IntCVSS vector, their final CVV vectors, exploitability, and

Table 7. Interaction vulnerability and impact metrics (CVV attributes)

Interaction	$IntCVSS$	CVV (as CVSS vector)	Exploitability	Impact	CVV
(Phone, sensor, P3)	P’, ’L’, ’N’, ’N’, ’U’, ’N’, ’H’, ’H’	’P’, ’L’, ’N’, ’N’, ’U’, ’N’, ’H’, ’H’	5.177088	3.887042775	9.064131
(Sensor, phone, C2)	A’, ’L’, ’L’, ’N’, ’C’, ’L’, ’L’, ’L’	A’, ’L’, ’L’, ’N’, ’C’, ’L’, ’L’, ’L’	2.068068156	3.73317227081746	5.80124
(Vendor DB, Phone, C4)	’N’, ’L’, ’N’, ’N’, ’C’, ’N’, ’N’, ’N’	N’, ’L’, ’N’, ’N’, ’U’, ’N’, ’N’, ’H	3.887042775	3.5952	7.482243
(Vendor OS, Vendor DB, C3)	’A’, ’L’, ’H’, ’N’, ’C’, ’H’, ’H’, ’L’	’A’, ’L’, ’H’, ’N’, ’C’, ’H’, ’H’, ’H’	0.900610326	6.04773049154452	6.948341
(Vendor OS, Vendor DB, C6)	’N’, ’L’, ’H’, ’N’, ’C’, ’H’, ’H’, ’L’	N’, ’L’, ’H’, ’N’, ’C’, ’H’, ’H’, ’H’	1.234707705	6.04773049154452	7.282438
(Azure OS, Azure compute, C3)	’A’, ’L’, ’H’, ’N’, ’C’, ’L’, ’L’, ’L’	’A’, ’L’, ’H’, ’N’, ’C’, ’H’, ’H’, ’H’	0.900610326	6.04773049154452	6.948341
(Vendor DB, Azure compute, C4)	’N’, ’L’, ’N’, ’N’, ’C’, ’N’, ’N’, ’N’	’N’, ’L’, ’N’, ’N’, ’U’, ’N’, ’L’, ’N’	3.887042775	1.4124	5.299443

Table 8. Initial impact values

	Sensor	Phone	AWS compute	Vendor DB	Vendor OS	Azure OS	Azure compute
Spoofing	2,50	2,50	2,00	2,00	2,50	2,50	2,00
Tampering	3,00	2,50	2,50	2,50	2,50	2,50	2,50
Repudiation	2,50	2,00	2,50	3,00	2,00	2,00	2,50
Information Disclosure	1,00	3,00	2,50	3,00	3,00	2,50	2,50
Denial of Service	3,00	3,00	2,50	3,00	2,50	2,50	2,50
Elevation of Privileges	3,00	3,00	2,50	3,00	2,50	2,50	2,50

impact values deviate. This occurs due to Equation (1), which combines the IntCVSS vector, the SingleCVSS vectors, and the ChainedCVSS vectors to output a single CVV per interaction. The initial IntCVSS vector for interaction `(VendorDB,AzureCompute,C4)` is `AV:N/AC:L/PR:N/UI:N/S:C/C:N/I:N/A:N`; the component has one cataloged vulnerability, so the ChainedCVSS and the SingleCVSS vectors are identical, namely `AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:H/A:N`. Combining these vectors through Equation (1), we can see that, in the final CVV vector `AV:N/AC:L/PR:N/UI:N/S:C/C:N/I:L/A:N`, the interaction has inherited part of the impact caused by the vulnerability. The high impact of the vulnerability, which would mean a total loss of integrity or protection, was not fully passed on to the interaction due to network controls mitigating part of the effect. The derived low integrity impact means that data modification might be possible from an attacker as per CVSS[16].

4.2.2. System risk analysis (supply chain)

As mentioned in Section 3, the system security analysis is quantified using STRIDE and DREAD methodologies to thoroughly identify the risk to each component and each threat. For each individual component, the initial impact and likelihood values associated with the STRIDE threats were computed by DREAD, as depicted in Tables 8 and 9. Damage represents the damage that a cyber attack may provoke to the systems; along with the affected users/systems, it represents the impact of the attack. Additionally, reproducibility represents the ability of the adversary to reproduce the attack, exploitability refers to the ability of the attacker to exploit the component’s vulnerabilities, and discoverability refers to the capacity of the adversary to identify system vulnerabilities. The sum of reproducibility, exploitability, and discoverability represents the likelihood of a cyber attack. Each column of Tables 8 and 9 represent individual components, indicated by their corresponding initials, as defined in Section 4.1. The values inside the cells are the corresponding impact and likelihood values per STRIDE threat and per individual component; these were calculated by Equations 2 and 3. Each of the DREAD variables - damage, reproducibility, exploitability, affected users/systems, and discoverability - accepts an integer value in [0,3], the value being assigned by considering the specific DREAD criteria described in[4].

4.2.3. Towards system security architecture (supply chain)

For each STRIDE threat, a set of security controls was selected by applying the methodology described in Section 3 to reduce the initial risk. The primary set of controls is listed in NIST guidelines for Cybersecurity Supply Chain Risk Management Practices[1] . In this report, cybersecurity supply chain risk management (C-SCRM) is integrated into risk management activities by applying a multilevel, C-SCRM-specific approach, including guidance on the development of C-SCRM strategy implementation plans, C-SCRM policies, C-SCRM plans, and risk assessments for products and services.[1] The first column represents the global initial risk that would

Table 9. Initial likelihood values

	Sensor	Phone	AWS compute	Vendor DB	Vendor OS	Azure OS	Azure compute
Spoofing	1,00	1,67	1,33	1,33	1,67	1,67	1,33
Tampering	2,00	2,00	1,67	2,00	1,33	1,33	1,67
Repudiation	1,33	1,67	2,00	2,00	2,00	2,00	2,00
Information disclosure	2,00	2,00	2,00	2,00	2,00	2,00	2,00
Denial of service	2,00	2,00	2,33	2,00	2,00	2,00	2,00
Elevation of privileges	1,67	1,33	2,00	2,67	2,00	2,00	2,00

occur without any security controls, as calculated in the risk analysis step. The second column represents the optimal set of security controls that reduce the risk, which is represented as the residual risk in the third column.

The optimal set of security controls towards the security architecture of the supply chain architecture depicted in Figure 4 are described in Tables 10-15. Particularly, the optimal set of security controls per STRIDE threats and per system are depicted along with the associated risks per threat. The security risks represent both the initial risk value - prior control selection - and the residual risk value - post control selection. *Denial of service* and *elevation of privileges* threats are characterized as the most critical threats for the targeted supply chain infrastructure. *Tampering*, *repudiation*, and *information disclosure* threats are characterized as medium level threats, while *spoofing* is a low level threat since the initial risk is the lowest with the value 1.34.

The security controls depicted in the tables below were identified by leveraging an automated decision support tool proposed in this work. Although the security controls reduce the initial risks of each component and each threat, their applicability is should be verified by domain experts and stakeholders together. The proposed methods enable the execution of what-if scenarios by modifying the initial list of the available security controls from NIST[1] and/or the parameters of the genetic algorithm.

By observing Tables 10-15, it is obvious that the optimal security controls for the reference supply chain management system consist of a large number of controls.

For the spoofing threat, the component that is in need for more security controls is the Vendor DB, as all collected data are stored there, and any successful spoofing attempt would result in a high-impact security incident, which could pose a significant risk to the entire system. For all other components of the system, it is also required to apply a number of spoofing-related security controls to limit the propagated risk for the system. For the tampering threat, the same holds for the Vendor DB because of the significance of the data stored therein. In general, it is observed that controls are significantly required in total to achieve similar risk reduction with the spoofing threat. For the repudiation threat, even fewer controls are required for the same risk reduction effect. For information disclosure, it should be noted that the phone must be more protected than other components (with three controls), as it seems that more vulnerabilities on the phone may be exploited for improper retrieval of information. For the denial of service threat, endpoint components such as sensors and phone produce low impact slightly on the system in general, and thus no controls are selected for them. The Azure components (OS and compute) are more vulnerable and need to be more protected to mitigate high risk propagation to the system. Finally, regarding the elevation of privileges threat, it is not highly critical to protect the endpoint components and the most prominent issues are identified for the AWSCompute component.

5. CONCLUSIONS

We present the combination of two existing methodologies that both deal with a risk analysis of cyber-physical systems. The proposed methodology is based on the fusion of two existing CPS-specific methodologies: the

Table 10. Optimal set of security controls per CPS component for STRIDE threat: spoofing

		Spoofing - optimal controls	
Initial risk	**Component**	**Control**	**Residual risk**
1.34	Sensor	Component authenticity (SR-11)	0.80
		Component authenticity anti-counterfeit training [SR-11(1)]	
	Phone	Account management (AC-1)	
		Identification and authentication policy and procedures (IA-1)	
		Authenticator management (IA-5)	
		Boundary protection (SC-7)	
	AWS compute	Security assessment and authorization policies and procedures(CA-1)	
		Identification and authentication (non-organizational users) (IA-8)	
		System maintenance policy and procedures (MA-1)	
		System and communications protection policy and procedures (SC-1)	
	Vendor DB	Access enforcement(AC-3)	
		Use of external information systems (AC-20)	
		Security assessment and authorization policies and procedures (CA-1)	
		Plan of action and milestones (CA-4)	
		Identifier management (IA-4)	
		Nonlocal maintenance (MA-4)	
		System security and privacy plans (PL-2)	
		Security categorization (RA-2)	
		System documentation (SA-5)	
		Malicious code protection (SI-3)	
		Security alerts advisories and directives (SI-5)	
		Software firmware and information integrity (SI-7)	
	Vendor OS	Contingency planning policy and procedures (CP-1)	
		Baseline selection (PL-10)	
		Risk assessment (RA-3)	
		Risk response (RA-7)	
	Azure OS	Contingency plan (CP-2)	
	Azure compute	Authorization (CA-5)	
		Identification and authentication (non-organizational users (IA-8)	
		Allocation of resources (SA-2)	
		Acquisition process (SA-4)	

Table 11. Optimal set of security controls per CPS component for STRIDE threat: tampering

		Tampering - optimal controls	
Initial risk	**Component**	**Control**	**Residual risk**
1.60	Sensor	Physical access authorizations (PE-2)	0.97
	Phone	Role-based security training (AT-3)	
		System and communications protection policy and procedures (SC-1)	
	AWS compute	Information exchange (CA-3)	
		Authorization (CA-5)	
	Vendor DB	Security assessment and authorization policies and procedures (CA-1)	
		Authenticator management (IA-5)	
		Identification and authentication (Non-organizational users) (IA-8)	
		Boundary protection (SC-7)	
		Inspection of systems or components (SR-10)	
	Vendor OS	Information system monitoring (SI-4)	
		Software firmware and information integrity (SI-7)	
	Azure OS	Personnel screening (PS-3)	
	Azure compute	Authenticator management (IA-5)	

first one is a methodology to assess IoT-enabled, cyber-physical attack paths[11], while the second one is a CPS methodology for risk propagation and control selection[19].

However, we choose to maintain a properly modified and integrated and carefully selected subset of the components of the previous methodologies to provide more efficient risk assessment results and verifiable risk treatment policies. To model complex CPS components and their relation, we use the method in[11], which allows capturing detailed cyber and physical interaction types among the system assets. The combined methodology allows for an accurate interaction modeling of the components of a CPS that outputs interactions for such components. The interaction assessment phase initially proposed in[11] is also selected to allow our new

Table 12. Optimal set of security controls per CPS component for STRIDE threat: repudiation

		Repudiation - optimal controls	
Initial risk	**Component**	**Control**	**Residual risk**
1.60	Sensor	Software firmware and information integrity	0.80
	Phone	Event logging (AU-2)	
	AWS compute	Event logging (AU-2)	
		Identification and authentication policy and procedures (IA-1)	
	Vendor DB	Identification and authentication (Non-organizational users) (IA-8)	
	Vendor OS	Malicious code protection (SI-3)	
	Azure OS	Information system component inventory (CM-8)	
	Azure compute	Identifier management (IA-4)	

Table 13. Optimal set of security controls per CPS component for STRIDE threat: information disclosure

		Information disclosure - optimal controls	
Initial risk	**Component**	**Control**	**Residual risk**
1.60	Sensor	System and communications Protection policy and procedures (SC-1)	1.17
	Phone	Access control for mobile devices (AC-19)	
		System and communications protection policy and procedures (SC-1)	
		Malicious code protection (SI-3)	
	AWS compute	Access control policy and procedures (AC-1)	
	Vendor DB	Control assessments (CA-2)	
	Vendor OS	Identification and authentication (Non-organizational users) (IA-8)	
	Azure OS	Publicly accessible content (AC-22)	
		Risk assessment (RA-3)	
	Azure compute	Account management (AC-2)	

Table 14. Optimal set of security controls per CPS component for STRIDE threat: denial of service

		Denial of service - optimal controls	
Initial risk	**Component**	**Control**	**Residual risk**
1.77	Sensor	System and communications protection policy and procedures	1.34
	Phone	Information exchange (CA-3)	
	AWS compute	-	
	Vendor DB	Control assessments (CA-2)	
	Vendor OS	Contingency plan (CP-2)	
	Azure OS	Wireless access (AC-18)	
		Publicly accessible content (AC-22)	
		Authenticator management (IA-5)	
	Azure compute	Impact analyses (CM-4)	
		Access restrictions for change (CM-5)	

Table 15. Optimal set of security controls per CPS component for STRIDE threat: elevation of privileges

		Elevation of privileges - optimal controls	
Initial risk	**Component**	**Control**	**Residual risk**
2.14	Sensor	-	1.24
	Phone	-	
	AWS compute	Information exchange (CA-3)	
		Authorization (CA-5)	
		System documentation (SA-5)	
		Security engineering principles (SA-8)	
	Vendor DB	Security assessment and authorization policies and procedures (CA-1)	
	Vendor OS	Remote access (AC-17)	
		Risk assessment policy and procedures (RA-1)	
	Azure OS	Access enforcement (AC-3)	
		Incident response plan (IR-8)	
	Azure compute	Use of external information systems (AC-20)	

methodology to take detailed vulnerability information in the form of CVEs as input.

Those interactions are then fed into the risk assessment and risk treatment modules that decide the optimal security control selection for a given system. For threat analysis, we chose the relevant module in[13], since it supports STRIDE analysis, which is a widely accepted model. The risk assessment phase of the proposed methodology is also based on a modified version of the risk analysis engine originally defined in[13], as it allows for the efficient computation of the overall system risk. The risk calculation is properly modified to take the vulnerability and impact assessment provided from the interaction assessment phase as input, which is not supported in the original risk analysis module in[13]. Another advantage of the proposed methodology with respect to the underlying methodologies[11,13] is the use of the attack path assessment functionality as a method for the validation of the control selection produced in the risk treatment phase. Attack paths may be optionally computed and assessed based on well-defined critical target systems. Then, by re-running the attack path assessment after considering the effect of the proposed security controls, it is possible to verify whether the selected controls may effectively mitigate the most critical attack paths.

As an initial validation, the proposed methodology was applied to a reference supply chain management system and the results obtained are extremely useful to operators of such systems. The set of optimal controls per component and per threat minimized the residual risk, as shown in Tables 10-15. For example, *the Azure Compute Identification and Authentication (Non-Organizational Users (IA-8)* control aims to deal with *the Azure Stack Spoofing Vulnerability (CVE-2019-1234)* of the Azure Compute component. Additionally, *the Information Exchange (CA-3)* control minimized the risk of the phone component that derives from *the MEMS ultrasound attack (CVE-2018-20823)* vulnerability.

Some extensions to the proposed system are being considered as future work directions. In this work, the CVV interactions [produced by Equation (1)] are used to define impact propagation in the risk propagation module. There exists an alternative approach that could utilize different CVV scores for impact and likelihood propagation, respectively. Another idea is to iterate the execution of the genetic algorithm with a different order for the STRIDE threats in order to identify which is the most cost-efficient combination of controls for all six threats. Finally, another point for consideration is to experimentally test the effect of the critical attack paths extracted from the attack path risk assessment module in enhancing the optimal control selection process.

DECLARATIONS

Authors' contributions

Wrote and reviewed the manuscript: Kavallieratos G, Grigoriadis C, Katsika A, Spathoulas G, Kotzanikolaou P, Katsikas S
All authors have contributed equally to the article.

Availability of data and materials

Not applicable.

Financial support and sponsorship

This research has been co-financed by the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH – CREATE – INNOVATE (Project ARTEMIS. Project code:T2EDK-02836). This work was funded in part by the Research Council of Norway, through the SFI ”Norwegian Centre for Cybersecurity in Critical Sectors (NORCICS)”, Project No.310105.

Conflicts of interest

All authors declared that there are no conflicts of interest.

Ethical approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Copyright

REFERENCES

1. Boyens J, Smith A, Bartol N, Winkler K, Holbrook A, Fallon M. Cybersecurity supply chain risk management practices for systems and organizations. Technical report, National Institute of Standards and Technology, 2022. Available from: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-161r1.pdf [Last accessed on 27 Oct 2022]
2. Humayed A, Lin J, Li F, Luo B. Cyber-physical systems security — a survey. *IEEE Int Things J* 2017;4:1802-31. DOI
3. Thakur K, Ali ML, Jiang N, Qiu M. Impact of cyber-attacks on critical infrastructure. In *2016 IEEE 2nd International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing (HPSC), and IEEE International Conference on Intelligent Data and Security (IDS)*, IEEE, 2016, 183–186. DOI
4. Kavallieratos G, Katsikas S. Attack path analysis for cyber physical systems. In: Katsikas S, Cuppens F, Cuppens N, Lambrinoudakis C, Kalloniatis C, Mylopoulos J, Antón A, Gritzalis S, Meng W, Furnell S, editors. Computer security. Cham: Springer International Publishing; 2020, pp. 19-33. DOI
5. Filho NG, Rego N, Claro J. Supply chain flows and stocks as entry points for cyber-risks. *Proc Comp Sci* 2021;181:261-8. DOI
6. Mensah P, Merkuryev Y. Developing a resilient supply chain. *Proc Soc behav sci* 2014;110:309-19. DOI
7. Warren M, Hutchinson W. Cyber attacks against supply chain management systems: a short note. *Int J Phys Distrib Logist Manag* 2000;30:710-6. DOI
8. Polatidis N, Pavlidis M, Mouratidis H. Cyber-attack path discovery in a dynamic supply chain maritime risk management system. *Comp Stand Interfaces* 2018;56:74-82. DOI
9. Ho W, Zheng T, Yildiz H, Talluri S. Supply chain risk management: a literature review. *Int J Prod Res* 2015;53:5031-69. DOI
10. ISO International Organization for Standardization. Iso 31000:2018 risk management - guidelines. 2018. Available from: https://www.iso.org/standard/65694.html [Last accessed on 27 Oct 2022]
11. Stellios I, Kotzanikolaou P, Grigoriadis C. Assessing iot enabled cyber-physical attack paths against critical systems. *Comp Secur* 2021;107:102316. DOI
12. Spathoulas G, Kavallieratos G, Katsikas S, Baiocco A. Attack path analysis and cost-efficient selection of cybersecurity controls for complex cyberphysical systems. In: Katsikas S, Lambrinoudakis C, Cuppens N, Mylopoulos J, Kalloniatis C, Meng W, Furnell S, Pallas F, Pohle J, Sasse MA, Abie H, Ranise S, Verderame L, Cambiaso E, Maestre Vidal J, Sotelo Monge MA, editors. Computer Security. ESORICS 2021 international workshops. Cham: Springer International Publishing; 2022. pp. 74-90. DOI
13. Kavallieratos G, Spathoulas G, Katsikas S. Cyber risk propagation and optimal selection of cybersecurity controls for complex cyberphysical systems. *Sensors* 2021;21:1691. DOI
14. Official common platform enumeration (cpe) dictionary. Available from: https://nvd.nist.gov/products/cpe [Last accessed on 14 Oct 2022].
15. MITRE. Common vulnerabilities and exposures (CVE). Available from: https://cve.mitre.org/ [Last accessed on 14 Oct 2022].
16. FIRST. Common vulnerability scoring system (CVSS). Available from: https://www.first.org/cvss/ [Last accessed on 14 Oct 2022].
17. Microsoft. Chapter 3 – threat modeling. 2010. Available from: https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ff648644(v=pandp.10)?redirectedfrom=MSDN [Last accessed on 14 Oct 2022].
18. Shostack A. Threat modeling: designing for security. John Wiley & Sons, 2014. Available from: https://www.wiley.com/en-us/Threat+Modeling%3A+Designing+for+Security-p-9781118809990#permission-section [Last accessed on 27 Oct 2022].
19. Kavallieratos G, Katsikas S, Gkioulos V. Cyber-attacks against the autonomous ship. In: Katsikas SK, Cuppens F, Cuppens N, Lambrinoudakis C, Antón A, Gritzalis S, Mylopoulos J, Kalloniatis C, editors. Computer security. Cham: Springer International Publishing; 2019. pp. 20-36. DOI
20. Seifert D, Reza H. A security analysis of cyber-physical systems architecture for healthcare. *Computers* 2016;5:27. DOI
21. Rafiullah Khan, Kieran McLaughlin, David Laverty, and Sakir Sezer. Stride-based threat modeling for cyber-physical systems. In *2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*, IEEE, 2017, pp 1–6. DOI
22. Goldberg DE. Genetic algorithms in search, optimization and machine learning. boston. usa. 1989. Available from: https://www.semanticscholar.org/paper/Genetic-Algorithms-in-Search-Optimization-and-Goldberg/2e62d1345b340d5fda3b092c460264b9543bc4b5 [Last accessed on 27 Oct 2022]
23. Blickle T, Thiele L. A comparison of selection schemes used in evolutionary algorithms. *Evolut Comp* 1996;4:361-94. DOI
24. Kott A, Ludwig J, Lange M. Assessing mission impact of cyberattacks: toward a model-driven paradigm. *IEEE Secur Priv* 2017;15:65-74. DOI
25. Lyu X, Ding Y, Yang S. Bayesian network based c2p risk assessment for cyber-physical systems. *IEEE Access* 2020;8:88506-17. DOI
26. Tantawy A, Abdelwahed S, Erradi A, Shaban K. Model-based risk assessment for cyber physical systems security. *Computers & Security* 2020;96:101864. DOI
27. Abie H, Balasingham I. Risk-based adaptive security for smart iot in ehealth. In *Proceedings of the 7th International Conference on Body*

Area Networks, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2012, pp. 269–275.

28. Seale K, McDonald J, Glisson W, Pardue H, Jacobs M. Meddevrisk: risk analysis methodology for networked medical devices. In *Proceedings of the 51st Hawaii International Conference on System Sciences*, 2018. DOI
29. Mokalled H, Pragliola C, Debertol D, Meda E, Zunino R. A comprehensive framework for the security risk management of cyber-physical systems. In: Flammini F, editor. Resilience of cyber-physical systems. Cham: Springer International Publishing; 2019. pp. 49-68. DOI
30. Rosado DG, Santos-olmo A, Sánchez LE, et al. Managing cybersecurity risks of cyber-physical systems: The marisma-cps pattern. *Comp Industry* 2022;142:103715. DOI
31. Sahay R, Meng W, Estay DS, Jensen CD, Barfod MB. Cybership-iot: a dynamic and adaptive sdn-based security policy enforcement framework for ships. *Future Gener Comp Syst* 2019;100:736-50. DOI
32. Orojloo H, Azgomi MA. A method for evaluating the consequence propagation of security attacks in cyber–physical systems. *Future Gener Comp Syst* 2017;67:57-71. DOI
33. Liu B, Qu G. Vlsi supply chain security risks and mitigation techniques: a survey. *Integration* 2016;55:438-48. DOI
34. Ghadge A, Weiß M, Caldwell ND, Wilding R. Managing cyber risk in supply chains: a review and research agenda. *Supply Chain Manag Int J* 2019;25:223-40. DOI
35. Timothy Kieras, Muhammad Junaid Farooq, and Quanyan Zhu. Modeling and assessment of iot supply chain security risks: the role of structural and parametric uncertainties. In *2020 IEEE Security and Privacy Workshops (SPW)*, IEEE, 2020, pp. 163–170. DOI
36. FIRST.Org. *Common vulnerability scoring system v3.1: user guide*, 2019. Available from: https://www.first.org/cvss/v3-1/cvss-v31-user-guide_r1.pdf [Last accessed on 27 Oct 2022]
37. Kavallieratos G, Katsikas S. Managing cyber security risks of the cyber-enabled ship. *J Mar Sci Engin* 2020;8:768. DOI
38. Rothlauf F. Optimization methods. Design of modern heuristics. Berlin: Springer Berlin Heidelberg; 2011. pp. 45-102. DOI