Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/cose

Computers & Security

Assessing IoT enabled cyber-physical attack paths against critical systems☆

*Ioannis Stellios**, *Panayiotis Kotzanikolaou*, *Christos Grigoriadis*

GR18534, 80 Karaoli Dimitriou, Piraeus, Greece

ARTICLE INFO

Article history:
Received 18 December 2020
Revised 19 March 2021
Accepted 29 April 2021
Available online 9 May 2021

MSC:
00-01
99-00

Keywords:
IoT
Attack paths
Risk assessment
CVSS
Threat modelling

ABSTRACT

Internet of Things (IoT) increase the interconnectivity and interoperability of systems in various critical sectors, such as industrial control, healthcare and smart transportation systems. At the same time, as IoT technologies enable systems to interact both in cyber and physical ways, they also act as enablers of complex attack paths against critical systems. In this paper we propose a novel risk-based methodology for identifying and assessing IoT-enabled attack paths against critical cyber-physical systems. While the majority of existing approaches focus on cyber system connectivity only, the proposed methodology models both cyber and physical interactions. In comparison to existing cyber physical approaches that grow exponentially, our approach is significantly more efficient, by utilizing an attack tree topology; the critical system is set as the root (target) of an attack tree that is recursively build, based on the identified cyber-physical system interactions. Our methodology uses well-known building blocks such Common Vulnerabilities and Exposures (CVE), Common Vulnerability Scoring System (CVSS) and threat modeling. Furthermore,we significantly reduce false positives by prioritizing the identified attack paths in a risk manner, which, in turn, can assist decision makers in effectively mitigating multi-hop attack paths. To validate our methodology, we developed a proof-of-concept implementation and tested it using a realistic scenario from the healthcare sector. Our results show that the proposed methodology can efficiently identify and assess hidden and/or underestimated cyber physical attack paths.

1. Introduction

Cyber-physical systems (CPS) play a key role in various sectors such as industrial control, energy, transportation and healthcare. Until recently CPS used to operate in isolated physical and network environments. But the adoption of IoT related technologies, such as interconnected sensors and actuators, have transformed these air-gaped CPS by enabling real-time remotely managed functionalities that reduce operational costs, increase productivity and quality control and allow the provision of new innovative services. As a side effect, this increased interoperability and interconnectivity has lead to a sharp increase of their attack surface, since remote adversaries may exploit a multitude of potential IoT-enabled attack paths (Stellios et al., 2018). Indeed, remotely managed IoT devices equipped with various cyber and physical interfaces create new attack capabilities, since they may act as bridges between different and presumably segregated networks and technologies and interact in unpredicted ways such as by ex-

☆ 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 code:T1EDK-01958).

* Corresponding author.
E-mail addresses: jstellios@unipi.gr (I. Stellios), pkotzani@unipi.gr (P. Kotzanikolaou), grigoriadis@unipi.gr (C. Grigoriadis).

https://doi.org/10.1016/j.cose.2021.102316
0167-4048/

tending, abusing or misusing their cyber-physical interactions (Ronen et al., 2016; Ronen and Shamir, 2016). Even worse, the lack of security certification in IoT increases the vulnerability surface and the available attach paths. Traditional risk assessment methodologies fail to capture this novel and continuously evolving threat landscape. Although they examine the network connectivity of the systems and may capture the risks deriving from cyber interconnections between systems, they are not concerned with 'non-conventional' cyber interactions that rely on the *physical proximity* of typical network interfaces. For example, different network interfaces connected in unrelated networks, may actually allow an adversary controlling one interface to affect the security of the other, by intercepting non-encrypted communication or jamming another device's interfaces that if they operate in the same frequency band, (e.g. ZigBee/WiFi at 2,4 GHz). In addition, they fail to capture threats such as those deriving from the physical capabilities of devices (e.g. physical proximity with moving systems/parts) or those deriving from wireless Input/Output interfaces, such as audio and visual interfaces (Ronen and Shamir, 2016).

Motivation. As demonstrated by recent real-world incidents and proof-of-concept attacks (Stellios et al., 2018), such non-traditional interactions may be exploited to realize actual cyber attacks. For example, a remotely controlled industrial robot with moving parts, may be misused by adversaries to physically damage a nearby critical system (Maggi et al., 2017); wireless IoT may be exploited to trigger jamming or injection attacks, against nearby networks (O'Flynn, 2011; Petit et al., 2015); line-of-sight interfaces and light emission devices (e.g. infrared or smart lights) may be abused to create covert channels to leak information from, otherwise, air-gaped systems (Fiebig et al., 2014; Guri and Bykhovsky, 2019; Petit et al., 2015; Ronen and Shamir, 2016). To address the limitations of traditional security assessment, recent studies have focused on modeling cyber-physical attack paths in IoT systems (Agadakos et al., 2017) or to assess the integration of IoT infrastructure in typical information systems (Dorsemaine et al., 2017). However, very few methodologies consider both cyber and physical attack paths. In addition, none of the existing methodologies takes into consideration the risk characteristics, which leads to an exponentially large number of potential attack paths without any risk prioritization. Such 'noisy' results may not provide a reliable input, for cost timely and cost effective risk mitigation.

Contribution. We propose a novel risk assessment methodology for assessing IoT-enabled, cyber-physical attack paths against critical systems. Our main contribution is the identification and assessment of attack paths that may combine both typical cyber interactions among systems, as well as physical interactions usually enabled by IoT systems. To identify attack path scenarios that are meaningful/profitable for adversaries, we follow an attack tree approach that is target oriented and source driven: each critical system is considered as a potential target node for adversaries. Then, based on the identified interactions, a recursive algorithm is used to construct all the potential cyber-physical attack paths towards the target node. The exploitability of those attack paths is assessed for various adversarial scenarios with respect to: (a) the exposure of the initial (source) node of each attack path against different adversaries, and (b) the cumulative vulnerability of all the interacting nodes within each attack path. This allows us to filter out those interactions that are not "mature enough" to be exploited by adversaries and thus to reduce the number of the generated assessed attack paths by focusing only on attack paths that are more likely to be exploited based on their current exposure status. Ultimately, by using a properly modified risk formula the risk of the identified attack paths against realistic threat agents is calculated, thus providing "ready-to-use" information for applying cost-efficient mitigation controls. By developing a proof-of-concept implementation and by testing a realistic scenario, we validate the proposed methodology and we demonstrate that it can effectively discover hidden and underestimated complex cyber-physical attack paths of high impact and risk.

Paper Structure. In Section 2 we review the related work, while in Section 3 we describe in detail the proposed methodology. In Section 4 we implement and validate the proposed methodology, based on a realistic proof-of-concept scenario in the medical sector. Section 5 concludes this paper.

2. Related work

Attack graphs and CVSS

Analyzing attacks via attack graphs or attack trees is an active research area for several years, see for example (Ammann et al., 2002; Ingols et al., 2009; Jajodia et al., 2005; Lallie et al., 2020; Phillips and Swiler, 1998). Some attack graph methodologies have utilized CVSS metrics to assess the vulnerabilities or security risk. Cheminod et al. (2009) describe how traditional air-gaped industrial systems can be attacked from remote locations (e.g. Internet) by taking advantage of complex interconnectivity schemes and existing vulnerabilities. Gallon and Bascou (2011) integrate CVSS and attack trees in order to compute the severity of a multi-stage, multi-host attacks. In particular, they utilize the method described in Sheyner et al. (2002) to construct network attack trees, which then combine with applicable CVEs in order to assess the cumulative damage on hosts and consequently on the applicable networks. Cheng et al. (2012) propose a metric-level aggregation of individual CVEs from three different aspects: Attack vector, authentication and access complexity. In Li et al. (2019) Li *et al* describe a quantitative model of attacks on Distribution Automation Systems based on CVSS and attack trees. In particular, they propose an attack probability quantification model based on attack trees, in which the maximum probability of each attack path is calculated by utilizing CVEs of each leaf (node). Lallie et al. (2020) present a thorough analysis of more than 180 attack graphs/trees in order to evaluate the effectiveness of such methods.

Risk assessment for IoT-enabled Cyber-Physical systems In Ge et al. (2017) a five phase framework was proposed. The researchers presented attack scenarios such as a sinkhole attack in a smart home environment. According to the authors the limitations of the methodology included the difficulty to depict all diverse connectivity paths, no-connectivity attack scenarios (e.g. Distributed-Denial-of-Service - DDoS) het-

erogeneity on communication protocols and static network topology. Dorsemaine et al. (2017) examine the potential risks to a legacy Information System (IS) and IoT infrastructure whereas Liu et al. (2012) develop a dynamic risk assessment methodology for IoT that incorporates features from an *Artificial Immune System*. Kott et al. (2017) propose an abstractive threat modeling that is focused on the challenges involved when modeling large scale, diverse and complex networks. In Agadakos et al. (2017) Agadakos *et al* propose a framework to model both cyber and physical interactions of IoT, in order to identify unexpected chains of events and thus the potential impact of the addition or removal of a device on an existing network. The proposed model was based on time transitions and states and was evaluated in a smart-home setting. According to Agadakos et al. (2017) the exponential increase of possible combinations limit the scaleability of their methodology. Another limitation is that the model can produce 'false-positives', since, it considers all input/output combinations possible without any security assessment.

Sequeiros et al. (2020) present related work on attack and threat modelling for IoT systems and cloud mobile applications whereas in Alhanahnah et al. (2020) the authors present *IotCom*, an approach to discover hidden threats. In particular, the researchers analyzed multi-app coordination threats that can trigger infinity activation loops or chain coordination events that can lead to race conditions and physical wear of a device. Via their platform they were able to perform static analysis of multiple IoT applications and detect several events of safety violations.

A vulnerability-based risk assessment regarding *edge computing* and IoT was presented in George and Thampi (2019). Authors proposed a multi-attacker multi-target graphical model that included attackers, targets, vulnerability relations in the network in order to assess the risk at the edge computing devices and apply the corresponding mitigation strategies. Ghazo et al. presented a tool for automatic attack graph generation for computer and SCADA networks (Al Ghazo et al., 2019). The authors tested their proposed algorithm in a water treatment system.

In summary, although the aforementioned risk-based attack graph methodologies succeed in identifying attack vectors resulting from cyber connectivity, they are not concerned with attack paths that also involve physical interactions. On the other hand, works that consider both the cyber and physical connectivity (Agadakos et al., 2017; Alhanahnah et al., 2020; Sequeiros et al., 2020) may produce an exponentially large number of attack paths including a large portion of false positives, since they do not follow a risk/threat prioritization process. Thus modeling and assessing complex IoT-enabled multi-level attacks in an efficient way is an open challenge.

3. The proposed methodology

Assessing the risk of complex cyber-physical attack paths against a critical target system first requires the identification of all existing *interactions*, both direct and indirect, of other systems with the target system as well as with themselves. According to RA standards, risk calculation can be defined based on five different risk class types, as defined in Zambon et al. (2011) and Gritzalis et al. (2018), which rely on threat, vulnerability and impact factors. In our methodology, risk calculation properly combines 'Type 1' with 'Type 4' risk classes as follows. In 'Type 1' methods (Gritzalis et al., 2018; Zambon et al., 2011) risk is analysed in relation to a threat and an asset, (or a group of similar assets). The calculation combines the likelihood of a threat, the (*combined*) vulnerability of the asset(s) involved, and the impact of the threat in the (group of) asset(s), as shown in Eq. (1):

$$\text{Risk(Threat, Asset)} = \text{Likelihood(Threat)} \otimes \text{Vuln(Threat, Asset)} \otimes \text{Impact (Threat, Asset)} \quad (1)$$

The operator $\otimes$ denotes a combination between the risk factors (this can be implemented through a discrete risk matrix). In 'Type 4' methods, risk is analysed with respect to an asset that has previously been categorized as critical. The risk in relation to a threat combines the vulnerability of the critical asset only and the potential impact of the threat against the critical asset, i.e.:

$$\text{Risk(Threat, Crit.Asset)} = \text{Vuln(Crit.Asset)} \otimes \text{Impact (Threat, Crit.Asset)} \quad (2)$$

Since our goal is to assess the risk of attack paths of interacting nodes towards a critical target, we properly combine Eqs. (1) and (2) as follows. Let $\mathcal{T}$ denote the critical target system and let $\mathcal{D}$ denote the set of all the assets (devices) in scope. Note that $\mathcal{D}$ contains both typical Information and Communication Technology (ICT) systems, as well as cyber-physical and IoT or IoT-enabled components that may be directly or indirectly interconnected with $\mathcal{T}$. Let $\mathcal{AP} = (d_n \rightarrow \cdots \rightarrow d_1 \rightarrow \mathcal{T})$, $d_i \in \mathcal{D}$ denote an attack path of interacting nodes, where the threat is triggered in node d_n and the actual target of the attack is the critical target $\mathcal{T}$. Then the risk for such and attack path is defined as follows:

$$Risk(Threat, \mathcal{AP}) = Likelihood(Threat, \mathcal{AP}) \otimes Vuln(Threat, \mathcal{AP}) \otimes Impact(Threat, \mathcal{T}) \quad (3)$$

The reason for combining Type 1 with Type 4 risk classes was to allow for fine-grained threat and vulnerability input from open sources (as supported by Type 1), and at the same time focus on the input of the critical target system (as supported by Type 4 risk formulas). Since the proposed methodology is source driven and target oriented, our goal is to assess the risk for various threat agents that may trigger an attack at the source node of an attack path, in order to eventually affect the critical target system. In our model, *Asset* is replaced by an attack path $\mathcal{AP}$ of multiple interacting assets, where the destination of the path is the critical target system $\mathcal{T}$. The impact is assessed based on the consequences of the critical target $\mathcal{T}$. This is reasonable since the ultimate goal of the adversary is to harm the critical asset; the other systems in the path are used in order to extend the attack vector. However, the likelihood and the vulnerability assessment take into consideration the whole attack path, since the adversary is expected to combine any capability having on the interacting node, in order to gradually exploit all vulnerabilities within an attack path. Obviously, the optimal adversarial strategy is to combine vulnerabilities found at the entry point system d_n with vul-

nerabilities found in the whole chain, to pivot (horizontally or laterally) to the ultimate target $\mathcal{T}$.

3.1. Building blocks: CVSS and CVE

Common Vulnerabilities and Exposures[1] (CVE), developed by MITRE, is a list of uniquely identifiable vulnerabilities, and is a 'de facto' standard for numerous software products. The Common Vulnerability Scoring System (CVSS) (Erdősi, 0000) is an open framework that incorporates risk characteristics to assess the severity of CVE software vulnerabilities. CVSS in its latest version consists of three metric groups: *Base Score, Temporal*, and *Environmental* metrics. The Base Score includes the Exploitability and the Impact Metrics. The Exploitability Metrics include: the Attack Vector (AV) with possible values (N)etwork, (A)djacent network, (L)ocal and (P)hysical; the Attack Complexity (AC) with values (L)ow or (H)igh; the Privileges Required (PR) with values (N)one, (L)ow or (H)igh; the User Interaction (UI) with values (N)one or (R)equired; and the Scope (S) with values (U)nchanged or (C)hanged. The Impact Metrics include the Confidentiality (C), the Integrity (I) and the Availability (A) impact, in the scale of (N)one, (L)ow or (H)igh. The Base Score produces a score ranging from 0 (lowest) to 10 (most severe). A CVSS vulnerability is represented as a vector string, a compressed textual representation of the values used to derive the score. The Base Score can be modified with Temporal and Environmental metrics, to fine-tune the vulnerability level. Temporal metrics contribute to the final score by taking into consideration the current state of the vulnerability, e.g. whether a full patch exists or not. The Environmental metrics modify the base score to each custom environment; for example the implementation of network-layer security controls, relevant to the particular vulnerability. Depending on the organization under assessment it may be possible to apply temporal and/or environmental metrics "en masse" for specific device/interaction types.

In our methodology we adopt and make use of the CVSS v3.1 scoring system and its notation, in order to assess the vulnerability of the interactions between the nodes for both cyber as well as physical interactions. The reader is referred to FIRST.Org (2019) for detailed analysis of CVSS.

3.2. Terminology and definitions

In order to assist the reader, we will first define the basic terminology. Then, before describing the methodology in detail, we provide a high-level description.

Interactions We define as an *Interaction* between two systems (nodes), called the *source* node, say x and the *destination* node y and we denote as (x, y, type) the directional action or 'influence' that x may cause to y, due to their proximity and/or connectivity. We define two categories of interactions (each having detailed types): *physical* and *cyber* interactions.

Cyber interactions They include all the actions that may be triggered by the source towards the destination node, due to their cyber connectivity. In order to model cyber interactions, we make use of two characteristics: the network connectivity level and the logical access level. Concerning the network connectivity level x and y may either reside to the same network or they may be connected via different network segments and/or technologies. Concerning the logical access of x to y we distinguish three access levels, none, low and high. None implies that x has no logical access at all at y; low corresponds to user-level access whereas high corresponds to privileged (e.g. root/admin) access. Table 1 summarizes the cyber interaction types.

Physical Interactions These include all the actions that may be triggered by x to y due to their physical proximity. The physical Attack Vector (`AV:P`) described in CVSS (FIRST.Org, 2019), is applied for Machine-to-Machine (M2M) interactions that are capable to physically reach each another. 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.

Type P1 describes cases where devices equipped with moving parts or moving capabilities (e.g. IoT-enabled industrial robotic arm, a robot vacuum cleaner) are in proximity with the target system. Adversaries may exploit network/software vulnerabilities of the device to extend their physical reach and cause physical damage and/or gain physical proximity with the target system (Maggi et al., 2017). Type P2 describes I/O proximity for specific interfaces (e.g. optical). Such I/O interfaces can be vulnerable against nearby adversaries. For example, line-of-sight interfaces such as optical sensors of collision avoidance systems may be abused by introducing artifacts (Petit et al., 2015). Other examples include the abuse of line-of-sight interfaces for creating covert channels to exfiltrate data (Guri and Bykhovsky, 2019; Ronen and Shamir, 2016). Furthermore, audio or video I/O interfaces have been proved to leak information as described in Assange (2017). Finally P3 types are based on the fact that it is possible to cause jamming or even integrity attacks, when wireless interfaces that operate on the same bandwidth (even if they are running different protocols) are physically in range (e.g. O'Flynn, 2011; Petit et al., 2015).

Attack Paths Let $\mathcal{T}$ denote the critical target system and let $\mathcal{D}$ denote the set of all the assets (devices) in scope. We define as an *Attack Path* against a target system $\mathcal{T}$ and we denote as $\mathcal{AP} = (d_n \rightarrow \cdots \rightarrow d_1 \rightarrow \mathcal{T})$, $d_i \in \mathcal{D}$ a chain of interactions, where the threat is triggered in node d_n (the entry-point system) and the actual target of the attack is the critical system $\mathcal{T}$. We stress out that for systems that directly interact with the target, we will examine both cyber and physical interactions, since any direct interaction may be exploited by the adversary to harm the target system. For systems that are indirectly connected with the target, we only model their cyber dependencies, since they may be utilized in order to extend the attack vector, by successively compromising a chain of interactions towards the target system.

Cumulative Vulnerability Vector of an Interaction: CVV$((x, y, \text{type}))$ This is a CVSS-like vector representing the *combined* vulnerability level of an interaction. It has a central role in our methodology and is described in Section 3.5.
Cumulative Vulnerability Vector of an Attack Path: CVV$(\mathcal{AP}, AV)$ Similarly, it denotes a CVSS-like vector that represents the combined vulnerability level of an $\mathcal{AP}$ consisting of sev-

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

Table 1 – Cyber interaction types: A cyber interaction ($x \rightarrow y$) may belong to type C1–C6, based on the connectivity and the logical access of x to y.

	Logical Access		
Connectivity	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 based on the proximity between devices. The implied capabilities of the source node on the target system may involve physical tampering, manipulation of I/O interfaces or manipulation of shared-band network interfaces.

Type	Description	Interface	Examples	Common attack patterns
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	Cause destruction/obstruction.
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	I/O suppression/manipulation (e.g. introduce artifacts in optical sensors). Side-channel attacks (covert channels for data exfiltration).
P3	**Networks' proximity** (*x* and *y* at *different* networks that are in range)	Different, but shared-band wireless interfaces	e.g 802.11.x and 802.15.x operate at 2.4 GHz	DoS (jamming) - Packet injection attacks.

eral interactions. Its computation is described in detail in Section 3.7.

3.3. A high-level description

The proposed methodology will utilize CVSS information in order to construct CVSS-like vectors that will enable the assessment of the exploitability of the identified interactions, and ultimately the vulnerability level of attack paths against adversaries. In this way we will assess the implied capabilities of the source to the target node resulting from their interaction, in order to exclude those interactions that are not "mature enough" to be exploited by adversaries. Then, by combining the validated interactions, we will generate and assess attack paths that are more likely to happen, based on the current exposure status of their interactions. The proposed methodology, shown in Fig. 1, consists of the following phases:

Phase 1 - Interaction modelling: The goal of this phase is to model all potential cyber and physical interactions between the target $\mathcal{T}$ and all the devices in $\mathcal{D}$, as well as between devices themselves. It combines information such as a device's I/O and network interfaces, moving parts and their active ranges, devices' physical location, available networks with their cyber/physical characteristics and logical/physical access rules, to construct lists of interactions.

Phase 2 - Interaction vulnerability assessment: The main goal in this phase is to assess all the potential interactions identified in Phase 1 by defining the cumulative vulnerability level of each interaction, based on existing CVEs per device as well as on environmental information. Essentially, this phase filters out those interactions that are not 'mature enough' to be exploited by potential adversaries in their current state.

Phase 3 - Attack Path Construction. The goal of this phase is to construct all the attack paths against the target system, by exhaustively combining all the assessed interaction tuples provided by Phase 2. Attack paths may vary in length, by involving one or more interactions.

Phase 4 - Attack Path Assessment: Finally all the attack paths defined is Phase 3 are assessed so as to calculate their risk level, based on a practical implementation of Eq. (3). For each attack path the CVV of each interaction tuple is combined with the vulnerabilities of the initial node and the characteristics of various adversaries, to calculate the risk level for various attack path scenarios. Fig. 2 presents a graphical representation of the first three phases. As shown, the first modeled and assessed based on the input information, in order to construct valid attack paths to eventually be assessed.

3.4. Phase 1: interaction modelling

During this phase we utilize all available information regarding device's physical characteristics, I/O interfaces and network connectivity, to construct all their cyber and physical interactions as defined in Section 3.2. Algorithm 1 implements interaction modeling. Let *PT* the input for physical topology related information, *NT* for network topology and *AR* for access

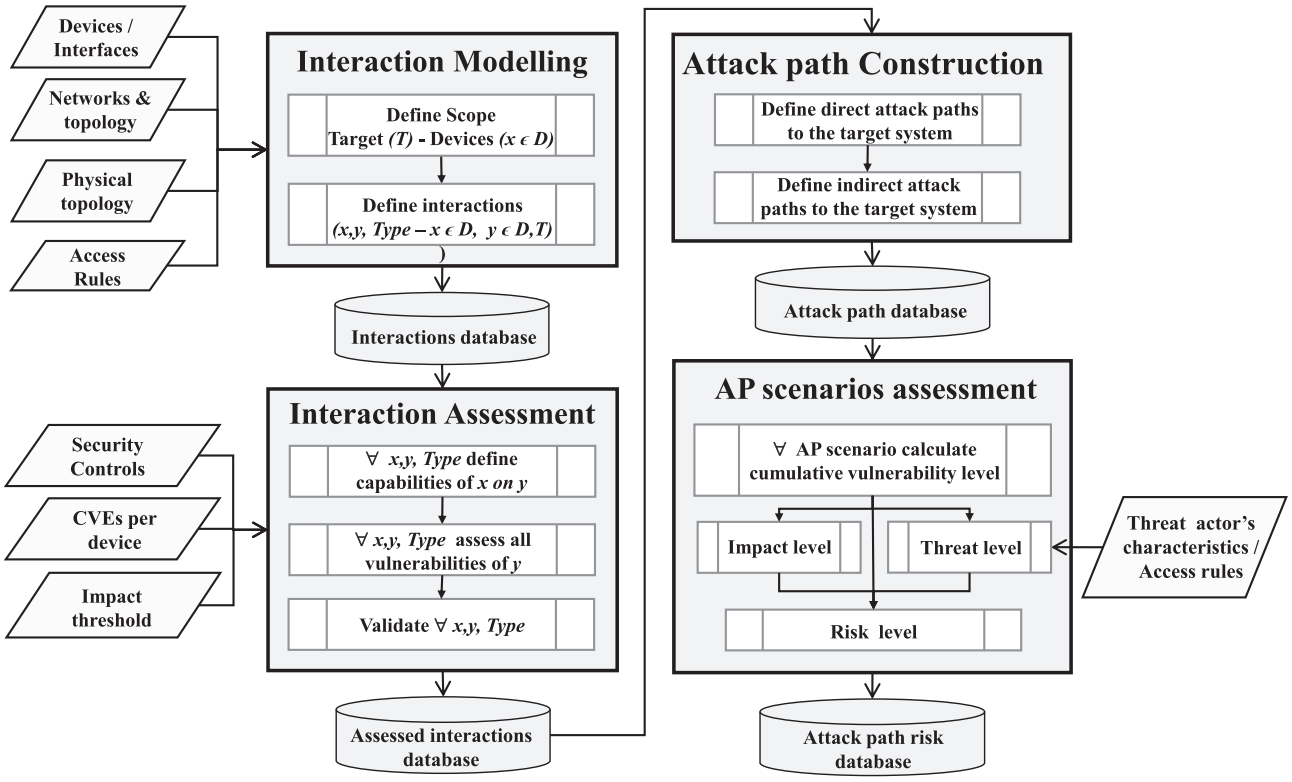

Fig. 1 – High level description of the proposed risk assessment methodology.

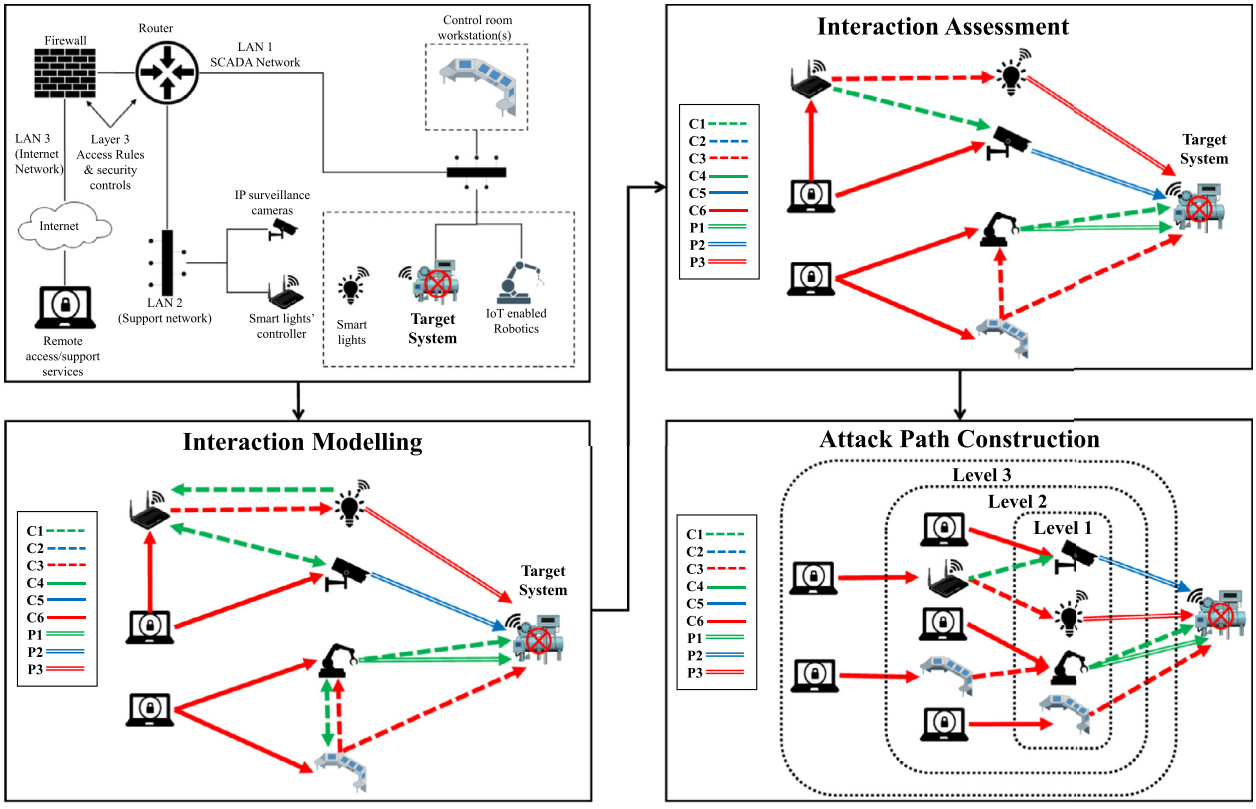

Fig. 2 – A graphical representation of a simplistic, yet realistic, risk assessment scenario that includes interaction modelling, assessment and attack path construction phases.

Algorithm 1: Identify and model all potential interactions in $\{\mathcal{D}, \mathcal{T}\}$.

Input : $\mathcal{T}$=Target system. $\mathcal{D}$=The set of devices in scope and their corresponding interfaces. PT=Physical Topology. NT=Network Topology. AR=Access Rules.

Output: $InteractionLists[]$ = A set of lists containing all direct interactions with the target system ($\equiv \mathbb{L}_1$) as well as the devices themselves ($\equiv \mathbb{L}_i, i = 2, 3...n$)

```
1  Algorithm ModelInteractions()
2      i ← 1                                   // Compute InteractionLists[1](≡ L_1)
3      InteractionLists[i] ← IdentifyInteractions(D, T, PT, NT, AR)
4      while (TRUE) do
5          InteractionLists[i + 1] ← ∅
           // Check all devices in Level-i as 'target' of any other
              device, in order to construct Level-(i+1) interactions
6          while ( (x, y, type) ← hasNext(InteractionLists[i]) ) do
7              L_x ← IdentifyInteractions(D, x, PT, NT, AR)      // x is a
                Level-i device
8              L_x ← L_x − (L_x ∩ InteractionLists[i])   // Don't duplicate in
                Level-(i+1), interactions already identified in
                Level-i.  Possible if graph has loops
9              InteractionLists[i + 1] ← InteractionLists[i + 1] + L_x
10         end
11         if (InteractionLists[i + 1] = ∅) then
12             break  // If no Level-(i+1) interactions exist, then exit
13         end
14         i ← i + 1
15     end
16 return (InteractionLists[]) /* Interaction lists for all existing
       levels (L_1, L_2, ..) */
```

capabilities respectively. The algorithm takes as input all of the described information, and outputs a set of lists (denoted as *InteractionLists*[]) containing all the direct and indirect interactions between the critical target $\mathcal{T}$ and any device $x \in \mathcal{D}$ in a structured way.

We define as the level-i interaction list (denoted as *InteractionLists*[i] = $\mathbb{L}_i$) the set of all the interactions of the form (x, y, type), where the shortest distance of the source node x from the target system $\mathcal{T}$ is *i* hops. In addition, since for the direct interactions we model both cyber and physical interactions, it holds that if (x, y, type) $\in \mathbb{L}_1$ then $y \equiv \mathcal{T}$ and type $\in$ [P1|P2|P3|C1| . . . |C6] (as defined in Tables 2 and 1). On the other hand, for all indirect interactions $\in \mathbb{L}_2, \ldots, \mathbb{L}_n$, it holds that $y \neq \mathcal{T}$ and type $\in$ [C1| . . . |C6].

Algorithm 1 works as follows. First, all the direct interactions with the target system are computed to form the list $\mathbb{L}_1$ (see lines 2–3 in Algorithm 1). Then, all the indirect interaction lists $\mathbb{L}_i, i = 2, \ldots, n$ are recursively computed, by exhaustively examining the potential interactions of all the source nodes in level-i interactions, but now as being destination nodes of possible interactions (lines 4–15). The algorithm avoids duplicating interactions already defined in previous lists, so that each interaction is defined once, in the shortest possible list. The procedure `IdentifyInteractions` is recursively called in the main algorithm. In the first call, since the destination of the interaction will be the target system $\mathcal{T}$, both physical and cyber interactions will be checked. For all other calls, only the cyber interactions will be modeled.

Since each call on `IdentifyInteractions` has computational cost proportional to $|\mathcal{D}|$, it is easy to see that the computational cost of Algorithm 1 will be proportional to $\mathcal{O}(|\mathcal{D}|^n)$ where *n* is the number of interaction lists. In our implementation, various ways are used to optimize the identification of interactions. First, during the identification of the systems

Algorithm 1: (continued) – Procedure *IdentifyInteractions*

```
/* Checks all devices x ∈ 𝒳 for interactions with device y. It
   outputs a list 𝕃 of all such interactions, described as vectors
   (x, y, Interaction_Type). */
IdentifyInteractions(𝒳, y, PT, NT, AR)
𝕃 ← ∅
while (x ← hasNext(𝒳)) do
    /* Check for cyber dependencies based on network connectivity,
       network topology and access rules */
    if ( C_i ← CyberInteraction(x, y, NT, AR)) ≠ ∅ then
        add(𝕃, (x, y, C_i))        // C_i ∈ (C1,...,C6) as defined in Table 1
    end
    /* Also check for physical dependencies with 𝒯, based on
       physical topology information PT (including the interfaces
       and physical capabilities of x) */
    if [ (y = 𝒯) and (P_i ← PhysicalInteraction(x, y, PT)) ≠ ∅ ] then
        add(𝕃, (x, y, P_i))        // P_i ∈ (P1,P2,P3) as defined in Table 2
    end
    𝒳 ← 𝒳 − x                  // Remove x and continue until 𝒳 is empty
end
return 𝕃
```

Algorithm 1 – Continued

$(\mathcal{D}, \mathcal{T})$ the physical characteristics (such as movement capabilities or proximity-based network interfaces and other non-typical interfaces) are identified. Thus, physical dependencies will only be examined for nodes with such capabilities. For example, a device equipped with moving parts/capabilities must be within its operating radius range in order to interact with the target system. Similarly, devices equipped with wireless interfaces are examined for physical interactions with the target system if their interfaces operate in the same frequency (see Appendix A, Table A.13). This information is assumed in *PT* in our algorithm. For the cyber interactions, during the identification of the nodes, each network interface of each node will be assigned to its corresponding network. Cyber interactions will then be identified based on network relations table as well as network access rules.

Furthermore, the logical access level for each interaction is examined, in order to define the level or remote access capabilities for each interaction tuple. This information is assumed in *AR* respectively.

3.5. *Phase 2: interaction assessment*

The goal of this phase is to filter out from further processing those interactions that are not 'mature enough' to be exploited by assessing their vulnerability level. For interactions that are considered as valid, their cumulative vulnerability level (CVV), as defined in Section 3.2, is calculated.

Assessing whether an interaction (*x*, *y*, type) is valid or not, is based on the level of the influence that *x* has on *y* due to their interaction. Recall that by definition, an interaction characterizes the influence that the source node *x* has on the destination *y*, due to their network connectivity or physical proximity. Assume that *x* has been compromised by the adversary (partially or fully). Then, the adversary can take advantage of all the capabilities of *x* on *y* in order to compromise *y* (partially or fully), as the next step towards the actual target system $\mathcal{T}$. Apart from the explicit access that *x* has on *y* due to their interaction, an adversary controlling *x* may also attempt to extend the control on *y*, by exploiting the existing vulnerabilities of *y*. For example, attempt to escalate the access level of *x* to *y* from user-level to admin-level access. *Assessment Strategy:* In order to assess an interaction, we first define their default (implied) impact and attack capabilities. These baseline attack capabilities of an interaction will be modelled using a CVSS-like vector, denoted as $IntCVSS_{base}$. Then, for each particular interaction we will use environmental information to transform the baseline capability interaction vector into a vulnerability vector, by taking into consideration the characteristics of the specific environment. The modified vulnerability interaction vector is denoted as $IntCVSS_{env}$. For example for a cyber interaction, $IntCVSS_{env}$ will take into consideration existing net-

Table 3 – Defining the implied capabilities for each of cyber interaction type as a CVSS vector.

		Exploitability Metrics					Impact Metrics		
	Type	AV	(M)AC	PR	UI	S*	(M)C	(M)I	(M)A
$IntCVSS_{base}$	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
$IntCVSS_{env}$		(M): These metrics can be environmentally modified (See Table 4) *Scope is unchanged (U), for level 1 interactions							

work security controls (if any), miss-configured access lists or context-specific access capabilities. For physical interactions, environmental information such as physical security controls and other context-specific information will be considered (e.g. Appendix A, Table A.15). Finally, in order to assess the possible ways that an adversary might exploit to escalate its control on y we will examine the resulting attack capabilities of x on y, with respect to the overall vulnerabilities identified on y.

3.5.1. *Defining the implied capabilities and impact of interactions*

As explained above, for each interaction type, we define a baseline CVSS-like vector, $IntCVSS_{base}$, representing the implied attack capabilities of x on y. *Cyber interactions* For the cyber interactions, recall that they have been defined based on the network connectivity and logical access of x to y (see Section 3.2 and Table 1). Thus, if x and y are connected at the same local network, we define the attack vector capability of the interaction as 'Adjacent Network' (`AV:A`), while for remote network connectivity, `AV:N` is assumed. Concerning the privileges required metric, we consider the implied logical access of each interaction type. In the case where the interaction type implies no access of x to y (e.g. nodes that only reside in the same/different network – types C1/C4), then we set the implied privileges of x to y to 'None' (PR:N). Similarly, the baseline privileges of x to y is set to 'Low', for types implying non-privileged access C2 and C5 (e.g. x is an aggregator that has limited capabilities of to reading/writing and/or execute data on a sensor y). Finally, for types C3 and C6 the privileges metrics are set to 'High' (e.g. x is an e-health web server that is able to remotely administer critical functions of an IoT-enabled medical infusion pump y). For the rest of the exploitability metrics we set the baseline attack complexity to 'High' and the user interaction to 'None'. The motivation is to assume as default values the most favorable for the adversary (although these values are modified when environmental characteristics are applied). Concerning the impact metrics, we consider that if an interaction does not imply any access privileges of x on y (C1 and C4), no impact can be caused on y by default. For interaction types that consider low level access of x on y (C2 and C5) we set the implied impact on y to "Low" for all impact metrics (C-I-A), proportionally to the impact of a user access vulnerability. Similarly, for C3 and C6, we set the implied impact to "High" for all impact metrics. Table 3 presents the $IntCVSS_{base}$ vectors for all cyber interaction types.

As presented in Table 4, the attack complexity and the impact metrics of $IntCVSS_{base}$ capability vector are modified in order to form the $IntCVSS_{env}$ vulnerability vector, depending on the available environmental information regarding security controls on network and/or application layer. For example, lack of security controls reduces the required `AC` of an $IntCVSS_{base}$ whereas a network security control (e.g. use of latest encryption schemes on network layer) can further reduce the corresponding impact metric (confidentiality).

Physical interactions For physical interactions we also use a similar approach. Due to physical proximity the attack vector is set as the implied access capability of x on y (`AV:P/A`) depending on the interaction type (see Section 3.2 and Table 2). For the rest of the exploitability metrics we follow a same reasoning as in the case of cyber interactions, allowing the most favorable metrics for an adversary as the default values. The only difference is that for all the types the implied privileges are set to "None", since a physical interaction does not require any kind of privileges (as defined in CVSS) of x on y. Finally we consider the scope as "Unchanged", since, physical interactions are only effect the target system $\mathcal{T}$. Similarly to cyber, the transformation of the baseline capabilities of physical interactions to a vulnerability vector is subject to environmental information. In particular, relevant security controls (see Appendix A table A.15) and the amount of damage that the source device's interface is capable of deliver to the target system are both taken into consideration for the final CVV to be calculated. Depending on the target type, several types of security controls may also be applicable (e.g. Force, 2017; Force and Initiative, 2013; Stouffer et al., 2011). An overview of how environmental security controls affect attack complexity and individual impact metrics of $IntCVSS_{base}$ capability vector is presented in Table 6.

3.5.2. *Identifying the vulnerabilities of the destination node*

For each target node of an interaction, we examine its existing vulnerabilities (CVEs). In addition, vulnerability chaining of single CVSS vectors is applied in specific cases, to assess the effect of *combined* vulnerabilities (see for example FIRST.Org, 2019). In any case, environmental information (temporal included) must first be applied before the vulnerability assessment and chaining process begins. *Single-vulnerability CVSS vectors* Depending on the cyber interaction type, CVEs can be considered as possible single (non-chained) vulnerability vectors, if their attack vector is adjacent or remote network

Table 4 – Proposed network environmental modifiers for $IntCVSS_{env}$ vector according to the corresponding security control level.

Network Security Controls	(M)AC	Impact Modifiers		
		M(C)	M(I)	M(A)
Not defined/Weak	H → L	No effect	No effect	No effect
Moderate	H	No effect	No effect	No effect
Strong	H	H → L	H → L	H → L
		L → N	L → N	L → N

Table 5 – Defining the implied capabilities for physical interactions as a CVSS-like vector.

		Exploitability Metrics					Impact Metrics		
	Type	AV	(M)AC	PR	(M)UI	S	(M)C	(M)I	(M)A
$IntCVSS_{base}$	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
$IntCVSS_{env}$		(M): Can be modified, based on physical environment (See Table 6.)							

Table 6 – Proposed physical environmental modifiers for $IntCVSS_{base}$ vector according to the corresponding security controls for each impact metric.

Physical Security Controls	(M) AC	Impact Modifiers		
		(M)C	M(I)	(M)A
Not defined/Weak	H → L	No effect	No effect	No effect
Moderate	H	No effect	No effect	No effect
Strong	H	H → L	H → L	H → L
		L → N	L → N	L → N

Table 7 – Summary of all vectors utilized in interaction assessment .

$IntCVSS_{base}$	A CVSS-like capability vector assigned on the interaction based on the interaction's type, using Table 3 (for cyber) or Table 5 (for physical interactions).
$IntCVSS_{env}$	The modified $IntCVSS_{base}$ vector based on environmental information for each particular interaction (e.g. see Tables 4 and).
$\{SingleCVSS\}$	A list of all the single CVSS vectors corresponding to vulnerabilities identified in y satisfying Eq. (6).
$\{ChainedCVSS\}$	A list of all the CVSS vectors of the chained vulnerabilities of y, computed based on Eq. (5) and satisfying Eq. (6).
$CVV((x, y, type))$	The Cumulative Vulnerability Vector of an interaction as defined on Eq. (7).

for C1-C3 (`AV:A/N`), or `AV:N` for C4-C6 respectively.

$$\forall\, CVE \texttt{ of } d \in \mathcal{D},\ \texttt{if AV:A/N then } CVE \in SingleCVSS \tag{4}$$

Chained-vulnerabilities CVSS vectors

Vulnerability chaining is based on the paradigm of FIRST.Org (2019) which demonstrates serial exploitation of vulnerabilities for privilege escalation, i.e. escalate the attack vector from local access to network or adjacent network (see Section 3.4 of FIRST.Org (2019)). In particular, we consider the cases where the exploitation of network vulnerabilities on y (`AV:A` or `AV:N`) that result in basic user access or an equivalent impact of `C:L/I:L/A:L` is combined with high-impact vulnerabilities (`AV:L`) to produce a chained vulnerability CVSS vector as described in Eq. (5)[2]:

$$ChainedCVSS = [AV : [N|A],\ max(AC),\ min(PR),\ max(UI),\ max(S),\ max(C, I, A)] \tag{5}$$

Validating CVSS vulnerability vectors After vulnerability chaining is complete, all of the identified (single and chained) vulnerabilities of y are examined, to verify which of them are exploitable based on the attack capabilities of x on y, as de-

[2] Function min/max is based on the following assumptions: `AV:N>A>L>P, AC:H>L, PR:H>L>N, UI:R>N, S:C>U, C/I/A:H>L>N.`

fined in $IntCVSS_{env}$. Eq. (6) is applied for each vector $CVSS \in \{SingleCVSS|ChainedCVSS\}$, *i.e.*

$$\text{If} \quad IntCVSS_{env}[Exploitability] \geq CVSS[Exploitability] \quad \text{then} \quad CVSS \in ValidCVSS \tag{6}$$

In Eq. (6) the operator $\geq$ has the following meaning for each exploitability metric: AV:A $\geq$ AV:N (i.e., if x is assumed to have adjacent network access to y, then it is capable to exploit vulnerabilities that require either adjacent or remote access); AC:H $\geq$ AC:L (i.e., if node x is capable to trigger attacks against y requiring high complexity, then it is also capable to trigger low complexity ones); PR:H $\geq$ PR:L $\geq$ PR:N (in the same sense is x is already assumed to have high privilege access on y then it will be able to also exploit vulnerabilities on y requiring low privilege access or no logical access at all). For the rest of the exploitability metrics the explanation is straightforward.

3.5.3. *Assessing the vulnerability level of the interaction*

Now the cummulative vulerability level of an interaction $CVV((x, y, \text{type}))$, defined in Section 3.2, is computed as follows. Recall that $IntCVSS_{env}$ is a cvss-like vector that defines the actual (environmental) capabilities that x has on y due to their interaction, and that *ValidCVSS* is a set of all valid (*i.e.* potentially exploitable) vulnerabilities identified on the destination node y, either single or chained ones. The vulnerability vector *CVV* that characterizes this interaction will be chosen among the above, based on the following procedure.

For all `level-1` interactions, the primary criterion for choosing the vector to be assigned as $CVV((x, \mathcal{T}, \text{type}))$ is considered the impact rather than the exploitability sub-score, since we are interested in identifying the maximum possible damage that the target node may exhibit by each interaction. For `level-i`, $i \geq 2$ interactions, the cumulative vulnerability level $CVV((x, y, \text{type}))$, $y \neq \mathcal{T}$, is assessed as follows. From the $IntCVSS_{env}$ as well as from all the valid single and chained vulnerability vectors of y, $CVSS \in ValidCVSS$ choose the one that: (i) concerning its impact metrics, it satisfies (C $\geq$ L&I $\geq$ L&A $\geq$ L) and (ii) has the highest exploitability sub-score. If more than one exist that satisfy the above criteria, choose the CVSS vector that has the maximum impact sub-score. The main motivation for this process is to ensure that interactions will be assigned to the *CVV* vector that corresponds to at least a partial compromisation on y (assured by the impact threshold) with the minimum required effort (i.e. the higher exploitability sub-score).

In both cases, if there exist more than one valid vulnerability vectors with identical exploitability and impact sub-scores, the single is preferred over the chained (if any). Finally, if no CVSS vector exists that satisfies the required criteria set in Eq. (7), the interaction is considered as invalid and *CVV* is set to $\emptyset$. These rules are described in Eq. (7). Note that the order of the arguments in function max denotes their priority in each case. Algorithm 2 summarizes the interaction vulnerability assessment phase.

$$CVV((x, y, \text{type})) = V \in (ValidCVSS_y, IntCVSS_{env}) \ \texttt{s.t.:} \begin{cases} \texttt{V has max(Impact,Exploitability)} & \text{if } y = \mathcal{T} \\ (C, I, A) \geq L \ \& \ \texttt{V has max(Expl., Impact)} & \text{if } y \neq \mathcal{T} \end{cases} \tag{7}$$

3.6. *Phase 3: attack path construction database*

In this phase all possible attack paths against the target system $\mathcal{T}$ are constructed, by exhaustively combining all the assessed interactions, produced in the previous phase. The attack path construction is described in Algorithm 3. The main algorithm (lines 1–22) works as follows. First, all the assessed level-1 interactions (i.e., direct interactions with the target system $\mathcal{T}$) are defined by default as one-hop attack paths ($\mathbb{AP}_1$). Then all the level-i attack paths $\mathbb{AP}_i$, $i > 1$, are computed recursively using $\mathbb{AP}_{i-1}$ and all the assessed interaction lists up to level-i ($\mathbb{AL}_1, \ldots, \mathbb{AL}_i$), by exhaustively examining if the destination node of a level-i interaction is the initial (source) node in each level-$(i-1)$ attack path. The final output is a list of lists $AttackPaths[i][j]$, containing all the valid chains of interactions of depth i towards the target system $\mathcal{T}$. The help procedures *isSource* and *append* are described for clarity.

Note that in Algorithm 3 the interaction tuples have been extended to also include their cumulative vulnerability vector, which was defined and assessed in Phase 2. In the case where interactions have null CVV value (recall that this is possible, as described in Section 3.5.3), they are considered as invalid and are excluded from any phase of the attack path construction (lines 7 and 12). It is easy to see that the computational cost of Algorithm 3 will be proportional to the product of the size of all the assessed lists, i.e., $\mathcal{O}(|\mathbb{AL}_1| \cdots |\mathbb{AL}_n|)$.

3.7. *Phase 4: attack path scenarios assessment*

The attack paths constructed in the previous phase can now be assessed. The risk of each attack path will be assessed using Eq. (3), as defined in Section 3. Recall that the risk for each attack path, takes into consideration the vulnerability of the whole attack path, the likelihood of a threat against the attack path being realized, and finally the impact on the actual critical target system.

The vulnerability level of each attack path combines the cumulative vulnerability level of all the interactions that form the attack path, i.e. $\{CVV\} \in \mathcal{AP}$, which have been assessed during the second phase (Section 3.5). In addition, we also consider the vulnerabilities of the initial ('entry') node of each attack path, i.e. the source node of the level-n interaction, for each attack path of length n. Recall that for each assessed interaction the *CVV* calculation has considered the capabilities of the source node and the vulnerabilities of the destination node. Thus, the vulnerabilities of the initial entry node have not been considered.

In order to examine all applicable threat agents against an attack path, for the initial node we first calculate all the applicable *CVV* vectors, one for each available `AV:N/A/L/P`). As in Section 3.5.3 each individual *CVV* must meet the impact threshold criterion. As defined in Section 3.2, we denote as $CVV(\mathcal{AP}, \texttt{AV})$ the *CVV* for a specific attack path and for a specific Attack Vector $\in$ [N|A|L|P]. For example $CVV(\mathcal{AP}_1, \text{N})$ denotes the CVV of $\mathcal{AP}_1$ for the attack vector "Network'. The threat level for each attack path will then be assessed based on threat modeling against each available `AV` of the initial node of the path. Recall that, by definition, this node will be the entry point for an adversary exploiting an attack path. Thus, we will model and assess all the applicable threat agents that are

Algorithm 2: Assess Identified Interactions (*AssessInteractions*)

Input : $InteractionLists[\,]\ (\equiv \mathbb{L}_i, i = 1, 2, ...n)$: A set of lists containing all interactions produced by Algorithm 1.
$\{CVE_d\}$: Sets of CVE/CVSS (environmental) vectors $\forall\ d \in \mathcal{D}$.

Output: $AssessedLists[\,]\ (\equiv \mathbb{AL}_i, i = 2, 3...n)$: A set of lists containing all assessed interactions.

```
AssessInteractions(InteractionLists[], {CVE_d})
for InteractionLists[i], i : 1 ... n do
    AssessedLists[i] ← ∅;   CVV ← ∅
    while ( (x, y, type) ← hasNext(InteractionLists[i]) ) do
        Define IntCVSS_base(x, y, type) /* Based on Tables 3,5 */
        IntCVSS_env(x, y, type) ← ApplyEnv(IntCVSS_base(x, y, type))
            /* As defined in Tables 4,6 */
        if type ∈ [C1, ··· C6] /* Chaining cyber interactions */
        then
            for CVE ∈ {CVE_y} do
                SingleCVSS_y ← SingleCVE(CVE) // Based on Eq.(4)
                ChainedCVSS_y ← ChainCVE(CVE) // Based on Eq.(5)
                ValidCVSS_y ← ValidCVE(SingleCVSS_y, ChainedCVSS_y)
                    // Based on Eq.(6)
            end
        end
        CVV ← CalcCVV(ValidCVSS_y, IntCVSS_env) /* Calculate
            interaction's CVV as described on Eq.(7) */
        add(AssessedLists[i], (x, y, type, CVV))
    end
end
return AssessedLists[i],   i = 1, ..., n
```

capable of utilizing different attack vectors against the initial node. For each attack vector of an attack path, the corresponding threat level is determined by taking into consideration the relevant CVV$(\mathcal{AP}, AV)$ exploitability metrics, physical/network characteristics of the initial node, as well as adversarial profiling features including, among others, required resources, motivation and even current threat landscape reports. Finally, the impact level for all attack paths will be based on the actual business impact that the loss of confidentiality, integrity and availability of the target system has on the organization. We utilize the impact metrics (C,I,A) of the level-1 interaction where $\mathcal{T}$ is the destination node, and modify them properly by applying the corresponding *Impact Subscore Modifier* as defined in the CVSS scoring system.

3.7.1. *Vuln(Threat, $\mathcal{AP}$): calculating the vulnerability level of attack paths*

As discussed above, for an attack path $\mathcal{AP}$, this process will combine the cumulative vulnerability CVV of each interaction involved in $\mathcal{AP}$ along with the vulnerabilities of the initial node of a path, to form, for each attack path, the Cumulative Vulnerability Vector(s) for all existing attack vectors of the path's entry node, i.e. CVV$(\mathcal{AP}, AV)$. At first all individual CVEs of the initial entry node are processed to form single and/or chained CVSS vectors.

Similarly to Section 3.5.2, for each possible AV a single or a chained vulnerability with the highest impact and exploitability sub-score is selected to from the CVSS vulnerability vector. Each of the latter is then combined using Eq. (8):

$$\text{CVV}(\mathcal{AP}, AV) = [\text{AV} : [\text{N}|\text{A}],\ \max(\text{AC}),\ \max(\text{PR}),\ \max(\text{UI}),\ \max(\text{S}),\ \text{Level}_1(\text{C}, \text{I}, \text{A})] \tag{8}$$

3.7.2. *Likelihood(Threat, $\mathcal{AP}$): calculating the threat level of attack paths*

After all the relevant CVV$(\mathcal{AP}, AV)$ have been calculated, the threat likelihood can be defined. In order to calculate the threat level one must first identify all available profiles of

Algorithm 3: Attack Path Construction Algorithm

Input : $AssessedLists[i] \equiv \mathbb{AL}_1, \ldots \mathbb{AL}_n$. A set of lists containing all the *assessed* interactions between devices themselves $\in \mathcal{D}$ (Level-2,...) and against the target system $\mathcal{T}$ (Level-1)

Output: $AttackPaths[i][] \equiv \mathbb{AP}_1, \mathbb{AP}_2, \ldots \mathbb{AP}_n$. A list of lists containing chains of interactions from an initial node $\in \mathcal{D}$ against $\mathcal{T}$. $\mathbb{AP}_i$ will contain the attack paths of depth i.

```
1  Algorithm ConstructAttackPaths()
2      for (i ← 1; i = n; i ← i + 1)   // Initialize all attack path lists.
        n:# of assessed lists
3      do
4          AttackPaths[i][] ← ∅
5      end
       // Define AP_1 first. By default, all interactions ∈ AL_1 are
          level-1 Attack Paths.
6      i ← 1,  j ← 1
7      while ( (x, y, Type, CVV) ← hasNext(AssessedLists[i]) and CVV≠ ∅ ) do
8          add(AttackPaths[i][j], [(x, y, Type, CVV)])
9          j ← j + 1
10     end
       // Recursively compute AP_i, i ∈ 2, ..., n using AP_{i−1} and AL_i.
11     i ← i + 1
12     while ( (x, y, Type, CVV) ← hasNext(AssessedLists[i]) and CVV≠ ∅ ) do
13         j ← 1,  k ← 1
14         while ( AttackPaths[i − 1][j] ← hasNext(AttackPaths[i − 1]) ) do
15             if (isSource(y, AttackPaths[i − 1][j]) ) then
16                 add(AttackPaths[i][k], append((x, y, Type, CVV), AttackPaths[i−
                    1][j]))
17                 k ← k + 1
18             end
19             j ← j + 1
20         end
21         i ← i + 1
22     end
23     return (AttackPaths[i][]) /* Attack paths AP_1, AP_2, .. */
```

threat agents that fit the organization under assessment. Then, the corresponding capabilities for each type of the adversary are defined by utilizing the CVSS exploitability metrics AV/AC/PR/UI (Fig. 3).

For example, a disgruntled employee is considered as someone with both logical as well as physical access to internal networks/devices (AV:N/A/L/P), restricted (user) access (PR:Low), basic computer skills (AC:Low) and is not relying on any user interaction in order to launch an attack (UI:None). On the other hand well-organized, groups of cyber criminals usually attack organizations from external networks (e.g. Internet - AV:N) without the need of any prior logical access (PR:None), consist of adversaries that are highly skilled (AC:High) and are capable of exploiting vulnerabilities with both UI:Required/None in order to gain initial foothold to the organization (e.g. via spear-phishing e-mails or by exploiting zero-day vulnerabilities). In order to define the

Algorithm 3: (cont.) – Procedures *isSource* & *append*

```
/* Boolean function that checks if a node d is the source node for
   a given attack path
   AttackPaths[i][j] = [(x_1, y_1, Type_1), ..., (x_i, y_i, Type_i)].  */
Procedure isSource(d, AttackPaths[i][j])
    (x_1, y_1, Type_1) ← AttackPaths[i][1]
    if (d = x_1) then
        return (TRUE)
    else
        return (FALSE)
    end

/* Takes as input an interaction and an attack path of depth i.
   Appends the given interaction at the beginning and returns a
   new attack path of depth i + 1.  */
Procedure append((x_0, y_0, Type_0), [(x_1, y_1, Type_1), ..., (x_i, y_i, Type_i)])
    for (k ← i; k=1; k ← k − 1) do
        (x_{k+1}, y_{k+1}, Type_{k+1}) ← (x_k, y_k, Type_k)
    end
    (x_1, y_1, Type_1) ← (x_0, y_0, Type_0)
    return([(x_1, y_1, Type_1), ..., (x_{i+1}, y_{i+i}, Type_{i+1})] )
```

Algorithm 3 – Continued

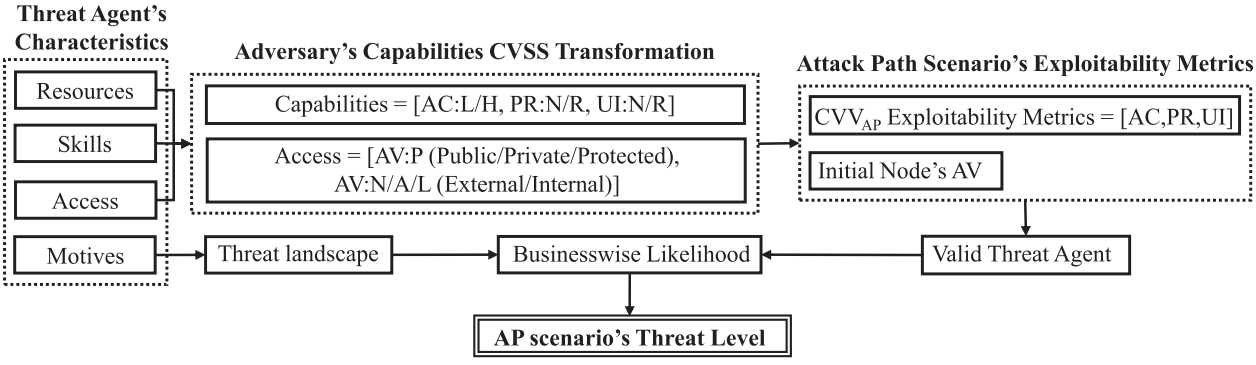

Fig. 3 – Threat level (likelihood) calculation methodology.

threat level we adopt the context and scale as described in Group et al. (2012).

For each AV of $CVV(\mathcal{AP}, AV)$ of the initial node (attack scenario) the corresponding access level (physical or network) of the adversary is defined. For physical access ($AV \equiv P$), *Public* applies to devices which are placed in a public places (e.g. an IP camera in a outside a building), *Private* can be considered an area where the access is limited to certain groups of people (e.g. an IP surveillance camera in a corporate garage accessible only to employees) whereas *Protected* can be considered a place heavily monitored and safeguarded by physical access security systems (e.g. a smart thermostat placed inside a data center). Similarly, for network access ($AV \equiv L, A, N$) we characterize as `internal` networks that are accessible from within the corporate environment whereas `external` are the ones that reside outside the organization's premises, Internet included.

In order to match all applicable threat agents for each attack path scenario each individual metric of $CVV(\mathcal{AP}, AV)$ is compared to the corresponding metrics of each attacker profile. Then, for the adversary types that satisfy all individual criteria described in the previous paragraph, the corresponding likelihood for each particular threat agent is applied.

Table 8 – Risk calculation matrix for assessing *Risk*(*Threat*, $\mathcal{AP}$) by combining *Vuln*(*Threat*, AP), *Likelihood*(*Threat*, $\mathcal{AP}$) and *Impact*(*Threat*, $\mathcal{T}$), as defined in Eq. (3).

Risk Level																								
Vulnerability Level	Impact Level																							
	Very Low					Low					Moderate					High					Very High				
	Threat Level																							
	VL	**L**	**M**	**H**	**VH**	**VL**	**L**	**M**	**H**	**VH**	**VL**	**L**	**M**	**H**	**VH**	**VL**	**L**	**M**	**H**	**VH**	**VL**	**L**	**M**	**H**	**VH**
Low	VL	VL	L	L	M	VL	L	L	M	M	L	L	M	M	M	L	M	M	M	M	M	M	M	M	H
Medium	VL	L	L	M	M	L	L	M	M	M	L	M	M	M	M	M	M	M	M	H	M	M	M	H	H
High	L	L	M	M	M	L	M	M	M	M	M	M	M	M	H	M	M	M	H	H	M	M	H	H	VH
Critical	L	M	M	M	M	M	M	M	M	H	M	M	M	H	H	M	M	H	H	VH	M	H	H	VH	VH

Risk Level: Very Low= VL, Low = L, Moderate =M, High = H, Very High = VH

3.7.3. *Impact*(*Threat*, $\mathcal{T}$): *calculating the impact level of attack paths*

In order to assess the actual impact that the organization suffers from each attack path in terms of CIA, we utilize each individual impact metric of the `Level-1` interaction tuple and apply the appropriate security requirement weights as defined in the CVSS scoring system. Guidelines for defining these weights according to the type of the target system can be found in the *CVSS Guide* (see §3.11 of FIRST.Org (2019)), as well as in several other publications such as Barker and NIST (2008). For example, the applicable security requirements" weights for a power generator could be set to `High` for integrity and availability and `Low` for confidentiality. In Table A.16 at the Appendix, we define the values of each individual CIA impact metric after the proper weight is applied. As in threat level, we adopt a [*Very Low*... *Very High*] scale, identical to the context and scale of NIST (Group et al., 2012). Finally the overall impact level can be computed by combining the individual (C,I,A) impact metrics (see Table A.17 in A).

3.7.4. *Attack path risk assessment*

By combining all the above information, the risk level of each attack vector for each attack path can be computed, according to our risk assessment formula of Eq. (3). This is essentially computed using the risk matrix shown in Table 8. Similarly to impact the context and scale of risk level is identical with the one described in Group et al. (2012).

3.7.5. *Attack path scenario risk mitigation*

Since the implementation of security controls varies, granular security policies can be tested and implemented, e.g. from applying low cost security controls like system patching, medium cost controls like ICT vulnerability patching, up to targeted policies such as software security hardening on the selected nodes.

Depending on a pre-selected risk threshold, the assessor can identify which attack path scenarios exhibit an unacceptable security risk. Then, the assessor can implement the mitigation plan based on the organization's security policies and procedures. In addition, our methodology enables the assessor to add alternative mitigation schemes. For example, if impact is considered of utmost importance the proposed strategy is to apply the appropriate security controls at all the nodes and corresponding networks of `level-1` interactions. In addition, the assessor may choose to eliminate certain types of adversaries just by focusing on applying the proper security countermeasures on entry nodes. Finally, in situations where security policies and procedures is difficult to implement and/or an intermediate response is needed, the assessor may choose to prioritize the mitigation process by selecting specific devices that have the highest multitude of attack path scenarios and/or are above a predefined risk level. All of the aforementioned mitigation scenarios can be simulated and the most efficient, cost beneficial security policies/procedures can then be selected.

4. Implementation and validation

In order to validate the proposed methodology a proof-of-concept implementation was created with in python3, utilizing several libraries. Pandas dataframes were used to structure and analyze the required input and output data of the application. The AST library was used in order to split complex input data from.csv files, so they can be inserted to lists and dataframes. For the vulnerabilities, the CVSS/CVSSlib library was used to calculate the base score (the exploitability and impact sub scores) of the interaction CVSS vectors and the newly produced CVSS vectors. The CVEs were collected from the NIST database and were pulled from the json files, based on their CPE identifier. For the implementation of Algorithm 1, the interaction tuples were properly adjusted and extended to also include the network id and the interface id used by the source and destination nodes. This extension aims to raise the complexity of attack paths from n^2 to $n^2 \cdot n_i$, where n is the number of devices and n_i the number of interfaces per device. During the interaction assessment phase (Algorithm 2), rules for network connectivity, physical interactions and security controls were applied. Then, capability vectors with the CVE/CVSS vectors of the destination node of each interaction, were utilized, along with python libraries CVSSlib/CVSS, for the calculation of the highest scoring vector for each `AV`, the CVV score and the production of the *Assessed Interaction Lists*.

The attack path construction module (Algorithm 3) is an iterative procedure that takes as input the *Assessed Interaction Lists*, along with an extensive CVSS centric rule-set, in order to produce a structured *Assessed Attack Paths Lists*. Finally for the attack path assessment, the CVV vector for each vulner-

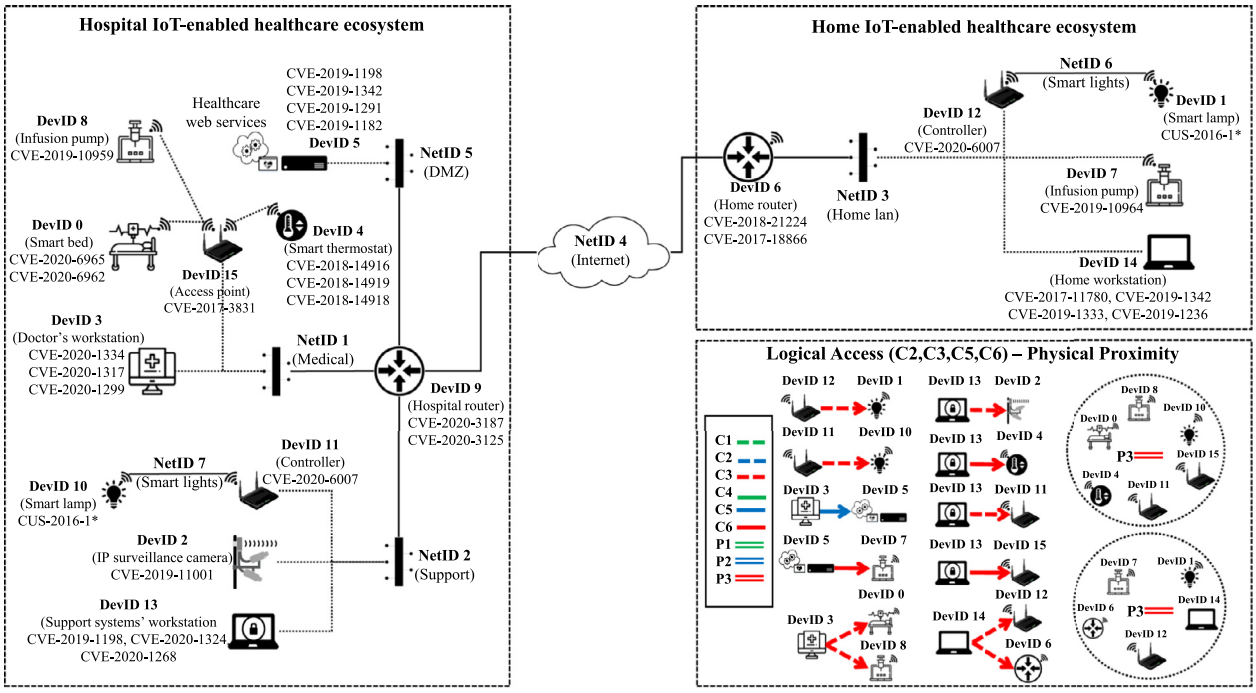

Fig. 4 – The simulated scenario - Network diagram, CVEs, cyber and physical proximity.

ability AV is calculated by utilizing CVSS/CVSSlib, based on the available vulnerabilities on the source node of each attack path. The exploitability metrics of the produced vector are then checked with each attacker's capabilities and the physical/logical access of the each adversary profile to the initial node.

4.1. *Test scenario*

To validate the efficiency and accuracy of the proposed methodology we used as a test case a realistic scenario from the healthcare sector (see Fig. 4). In particular, we focused on critical systems and services such as on-line remote healthcare services and near-patient infusion pumps. We simulated scenarios where the infusion pump is placed both in a smart home, as well as within the hospital. In addition, we included various low-importance IoT devices in both environments such as smart lamps, thermostats and IP surveillance cameras, as well as traditional ICT systems such as personal computers, network routers and access points. We defined logical access rules among the devices (e.g. to allow a doctor to monitor and reprogram infusion pumps via e-health services). In addition, for each device several well-known CVEs, or in some cases custom CVEs based on previous research were assigned (e.g. CUS-2016-1 Ronen et al. (2016)).

In order to be as realistic as possible we included popular medical devices and ICT equipment. In particular, we utilized two infusion pumps, one by 'BD Alaris' (near-patient, home) and another one by 'Medtronic' (in-hospital), as well as a patient monitor (Carescape B450 by 'GE healthcare'). In addition, we added IoT devices such as smart lighting systems (Philips), a smart thermostat, an IP-enabled surveillance camera with infrared interface as well as windows server(s) running remote medical services, network equipment by Cisco and D-Link and home/hospital workstations running Windows 10. For each device specific software version(s) based on the Common Platform Enumeration (CPE) standard (CPEIDs - see Table), corresponding vulnerabilities, cyber-physical interfaces, physical location (hospital/home), the corresponding network for each device's interface and logical access to other devices was assigned. Furthermore, we defined network relations (see Table), network access rules among devices as well as relevant security controls on network layer (environmental information). For each interface type we defined the corresponding cyber and physical interaction types, range and type (internal, external). For example, an infusion pump is physical located at the hospital (internal), has one wireless interface that communicates via a 802.11.x network (NetID 1 - internal), can interact with other devices with interfaces that operate in the same band (e.g. Philips hue smart lamps - interaction type P3) and is remotely managed by e-healthcare software (DevID 5). Except from traditional cyber attack vectors (AV:N/A/L) we also included non-traditional attack methods such as those described in Guri and Bykhovsky (2019).

Healthcare is an attractive sector for adversaries such as organized cyber crime, due to the great value of proprietary research data (e.g. COVID-19 vaccine) as well as patient's medical information such as Electronic Health Records (EHR) in the black market whereas healthcare organizations such as hospitals are considered as 'profitable business' of ransomware campaigns. In addition, COVID-19 pandemic increased the need for telehealth services and therefore the interest in dark web mentions increased 144% according to a recent threat report (SecurityScorecard, 2020). In our threat analysis we considered several types of realistic threat agents, ranging from highly motivated adversaries such as cyber criminals, to in-

Table 9 – Adversarial model for healthcare ecosystem (PoC).

Adversaries	Capabilities	Physical/Network Access Level	Motives	Resources	Likelihood
Healthcare Rights Activist	AV:N/AC:L/PR:N/UI:N	External	1	Limited	Low
Disgruntled Healthcare Worker	AV:N,A,L/AC:L/PR:N,L/UI:N,R	Internal (Hospital)	1,2	Limited	Low
Disgruntled Healthcare Systems' Administrator	AV:N,A,L,P/AC:H/PR:N,L,H/UI:N,R	Internal/Protected (Hospital)	1,2	Moderate	Low
Business Competitor	AV:N/AC:L/PR:N/UI:N,R	External(Internet)	1	Significant	Moderate
Cyber Criminals	AV:N/AC:L,H/PR:N/UI:N,R	External (Internet)	3,4,5	High	Very High/Low
Cyber Terrorist	AV:N,A,L,P/AC:L,H/PR:N/UI:N,R	External/Internal (Hospital/Home)	1,2,4	High	Moderate/Low
Nation State	AV:N,A,L,P/AC:L,H/PR:N/UI:N,R	External/Internal (Hospital/Home)	1,2,4,5	Very High	Low

Motivation: 1=Harm Reputation, 2=Damage/Disable equipment, 3=Financial Gain, 4=Harm Patient(s), 5=Steal Patients' Data (*)Likelihood: Hospital/Home

ternal, moderately motivated/skilled disgruntled employees. In addition, we defined specific motives known to be applicable to the healthcare sector. Finally, for each motive we took into consideration past and present threat reports ((for Network and, ENISA; Proofpoint, 2019; ProofPoint, 2020; SecurityScorecard, 2020)) including recent reported incidents (e.g. Scroxton, 2020) concerning the healthcare ecosystem in order to define the likelihood of each adversary type. We also applied different likelihood levels for same adversary types depending on the point-of-entry devices' environment (home/hospital). To test our methodology, we first identify all the cyber and physical interactions, using all devices in scope as possible targets. Then we calculated the attack paths for the three critical target systems: two medical pumps, one inside the hospital and the other in the home environment (DevIDs 7, 8) and also an e-health services web server (DevID 5). We assess the relevant interactions and we calculated the risk of the attack paths towards all the three predefined targets. In the attack path assessment phase we firstly computed all applicable CVVs of the initial node of each attack path and then we went on calculating the $CVV(\mathcal{AP}, AV)$ for each attack path scenario. In order to define the applicable threat agents for each attack path we compare their characteristics shown in Table 9 with each of the $CVV(\mathcal{AP}, AV)$ exploitability metrics.

To calculate the impact level for each attack path we utilize the vulnerability impact metrics of the 'Level-1' interaction of each attack path and apply the security requirement weights. In particular we defined the latter as `C:L, I:M, A:M/C:M, I:H, A:H` for in-home/in-hospital infusion pumps and `C:H, I:H, A:H` for the e-health services. In particular, we consider the impact of exploiting a single infusion pump placed in a home environment to be significant lower than the one of multiple infusion pumps installed in a hospital whereas the e-health web services is considered as a high impact target.

Finally, we utilized Table 8 from Section 3.7.4 to define the risk level of each attack path scenario.

4.2. *Results analysis*

In order to test the performance of the algorithm we run the simulation for the creation of interaction tuples using each node as the target device. Table 10 sums up the required time for computing all possible interactions. Then, we proceeded with the implementation of all of the methodology phases for the three selected critical targets.

4.2.1. *Interaction modelling and attack path construction phase*

From Table 11 we can infer that our target-oriented approach reduced the multitude of potential interactions of all devices, networks and interfaces for all three targets to 245 cyber-physical interaction tuples in total whereas in the vulnerability assessment of the interaction tuples phase the overall number was further reduced by 27% (182). From the latter 2103 attack paths were formed, of which 272 cyber-physical, for all three targets. Finally, for all the predefined threat agents, 6163 cyber and 1016 cyber-physical attack path scenarios (mappings of attack paths to applicable threat agents) were formed and assessed.

4.2.2. *Risk assessment phase*

Risk analysis of the formed attack path scenarios resulted in a variety of risk levels ranging from `Very Low (VL)` to `Very High (VH)` (see Fig. 5). In particular, 75 (1,2%) of the assessed cyber threat scenarios were characterized as `Very High` whereas the highest risk level of cyber-physical was `High` (4%).

The adversary risk profiles for the healthcare ecosystem paradigm is depicted in Fig. 6. By further analyzing the results we defined the AP scenarios that each device participated either as an intermediate node in the attack chain or as a Point-of-Entry. As shown in Table 12 the devices with IDs 3, 0 and 13 are the top three devices that are part of, or act as enablers for an AP scenario. In addition, the aforementioned devices were also the ones with the highest score concerning AP scenarios with risk levels `Very High` or `High`.

Besides the analysis and ranking of the attack paths and the relevant scenarios, and beyond the 'expected' high-risk paths, our methodology may assist the risk assessor to identify underestimated and/or hidden attack paths. We analyse three characteristic AP scenarios provided by our tool (see Fig. 7). We deliberately included high impact - low probability scenarios, as those are likely to be overlooked by typical risk assessment methodologies. The first is a stealthy, cyber-physical AP Scenario of high risk. A cyber-criminal takes

Table 10 – Interaction modelling calculation time (per target device/total/average).

Target Device	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	sum	averg
Time (sec)	1,71	1,46	0,80	1,02	1,04	1,14	1,34	1,40	1,11	0,70	1,19	0,85	1,39	0,84	1,39	1,01	1840	1,15
Levels	3	6	4	3	3	4	6	5	3	3	4	4	6	4	6	3	N/A	4,19
Interactions	113	142	109	108	76	118	97	75	113	107	124	112	137	109	140	99	1773	12,006

Table 11 – Interactions, attack paths and attack path scenarios per interaction level for all three targets.

	Level 1	Level 2	Level 3	Level 4	Level 5	Level 6
Interactions	23 (9 Phy)	87	87	50	1	0
Assessed Interactions	19 (9 Phy)	65	47	50	1	0
Attack Paths (Cyber)	10	47	154	454	688	478
Attack Paths (Cyber-Physical)	8	24	68	171	1	0
AP Scenarios (Cyber)	46	162	514	1555	2283	1603
AP Scenarios (Cyber-Physical)	16	66	246	682	6	0

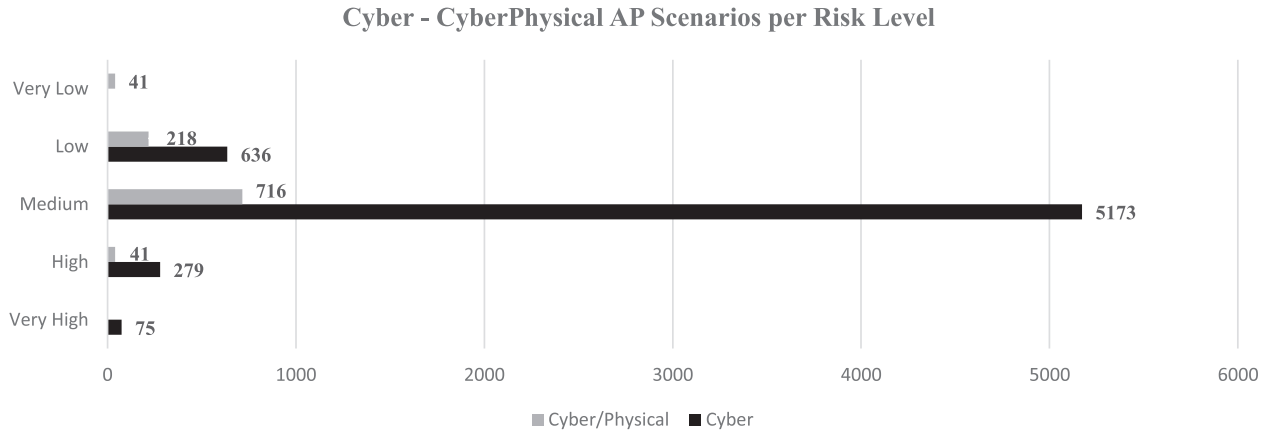

Fig. 5 – Cyber and cyber-physical attack paths scenarios per risk level.

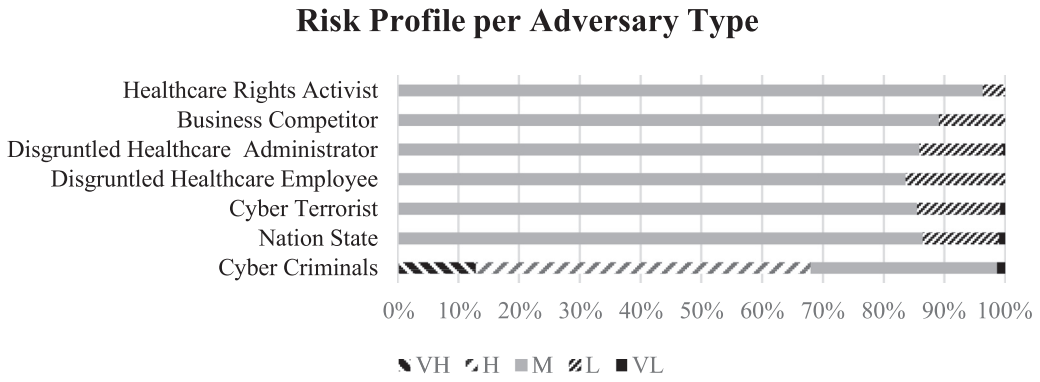

Fig. 6 – Risk profile of each predefined threat agent.

Table 12 – Multitude of AP scenarios per node for targetIDs 5, 7 and 8.

TargetID	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
AP Scenarios	4857	32	4574	4927	709	2458	122	0	2002	2568	288	3199	101	4742	92	2138
As point-of-entry	315	11	762	1016	0	9	24	0	465	2568	230	423	11	562	24	759

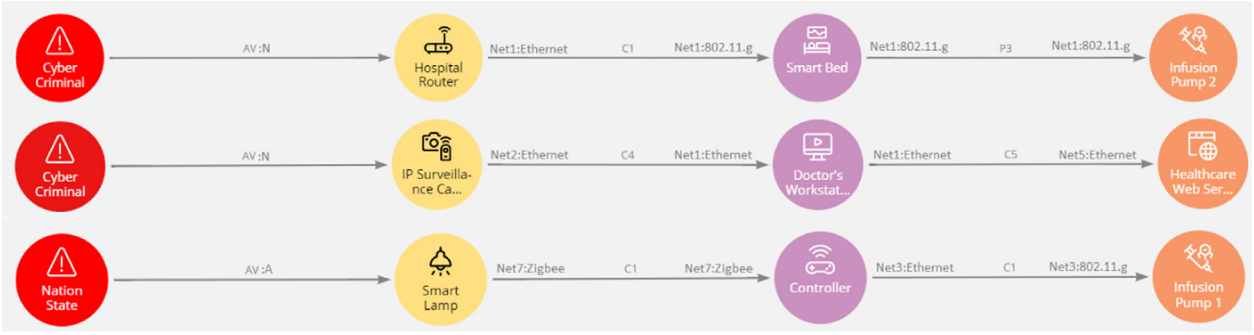

Fig. 7 – High impact, IoT-enabled, stealthy cyber/cyber-physical AP scenarios paradigms from our test scenario.

advantage of software vulnerabilities on the hospital's main router to gain initial foothold, then exploits vulnerabilities found on IoMT devices (patient healthcare monitors - smart beds) and causes DoS to multiple IoT-enabled infusion pumps by exploiting the physical proximity of interfaces working in the same band frequency). Such an attack path could be part of a ransomware campaign.

In the second AP scenario, a remote adversary exploits a critical vulnerability found on an Internet exposed IP surveillance camera; then exploits via lateral movement an IoT-enabled healthcare monitor to ultimately to gain access to hospital's web services and exfiltrate sensitive patient data. Finally, the third AP scenario is considered as a stealthy, high impact - low likelihood scenario. An highly skilled/resourced adversary (e.g. nation state) targets a home patient and via war driving techniques manages to infiltrate the patient's home network by exploiting vulnerabilities found in smart light bulbs and the corresponding controller. Then, she pivots into the home patient's network and takes advantage vulnerabilities found on the actual target (IoT-enabled infusion pump) ultimately threatening the patient's life.

4.2.3. *Risk mitigation*

After calculating the risk for cyber and cyber-physical AP scenarios we proceeded to the risk mitigation phase. In particular, we simulated a typical patch scenario which an organization would most likely implement in order to mitigate the risks. As the first step in a typical threat remediation process is to address the vulnerabilities found at the critical devices (targets). Then, the next stage is to patch the ICT equipment such as servers, workstations and crucial network equipment. Finally, the last step is considered to be addressing the vulnerabilities found on IoT devices. As depicted in Fig. 8 after patching all three critical systems there was a significant reduction from a total of 7179 to 4984 AP scenarios (31%). Especially for cyber AP scenarios there was a significant reduction (100% for `Very High`, 25% for `High`, 35% for `Moderate` and `Low`) whereas there was no reduction to cyber-physical ones, since, physical interactions with the target system do not rely on software vulnerabilities. In the next stage (ICT patch process) AP scenarios related with high risk level were fully mitigated, leading to a significant reduction from 4984 to just 95 (just 6 cyber and 95 cyber-physical) AP scenarios. The numbers of cyber-physical AP scenarios where further reduced to just 10 after IoT devices' vulnerabilities were addressed. The residual risks after the whole patching process was completed, where mainly due to insider threats such as adversaries with logical access to specific devices. (*Disgruntled Healthcare Systems' Administrator/ Worker*). All of the above are depicted in Fig. 8.

In order to further improve the mitigation process we strategically utilize the available information from the risk assessment phase. In particular, from the available information in Table 12 we classified the devices based on the multitude of AP scenarios (either as intermediate or entry node). We chose the first three devices with the highest score (IDs 3,0 and 13) and applied all security patches. We then run the simulation and discover a total reduction of 94% (from 7179 to just 396) for all AP scenarios (97% for cyber and 78,5% for cyber-physical). This in turn makes this approach a far more efficient way to reduce risks especially when a quick response is of utmost importance. In addition, an assessor may utilize the available information to prioritize the security countermeasures against specific types of adversaries and/or specific types of cyber or physical interactions (e.g. cyber criminals able to physically interact with a near-patient medical device from a remote location such as the Internet).

5. Conclusions

In order to address the risk deriving from the increased cyber and physical interoperability among ICT and IoT systems, we have proposed a target-oriented and source-driven methodology, in order to efficiently define and assess the attack paths against critical targets. By extending CVSS metrics we model and assess both cyber and physical interactions using vulnerability vectors, which are then utilized to assess various attack path scenarios. As demonstrated in our test scenario, our approach greatly reduces the number of identified paths, which may provide useful input for risk mitigation. Furthermore, by integrating exploitability metrics to threat agents' characteristics, we automated the risk assessment process in order to be able to identify and assess hidden realistic attack scenarios.

Future work includes the enrichment of interaction modelling phase by including additional physical interaction types. In addition, we aim to automate the interaction identification phase, by creating a cyber security ontology expressed as a knowledge graph that will improve the processing of temporal and environmental information provided by automated network scanning tools, to automatically produce network in-

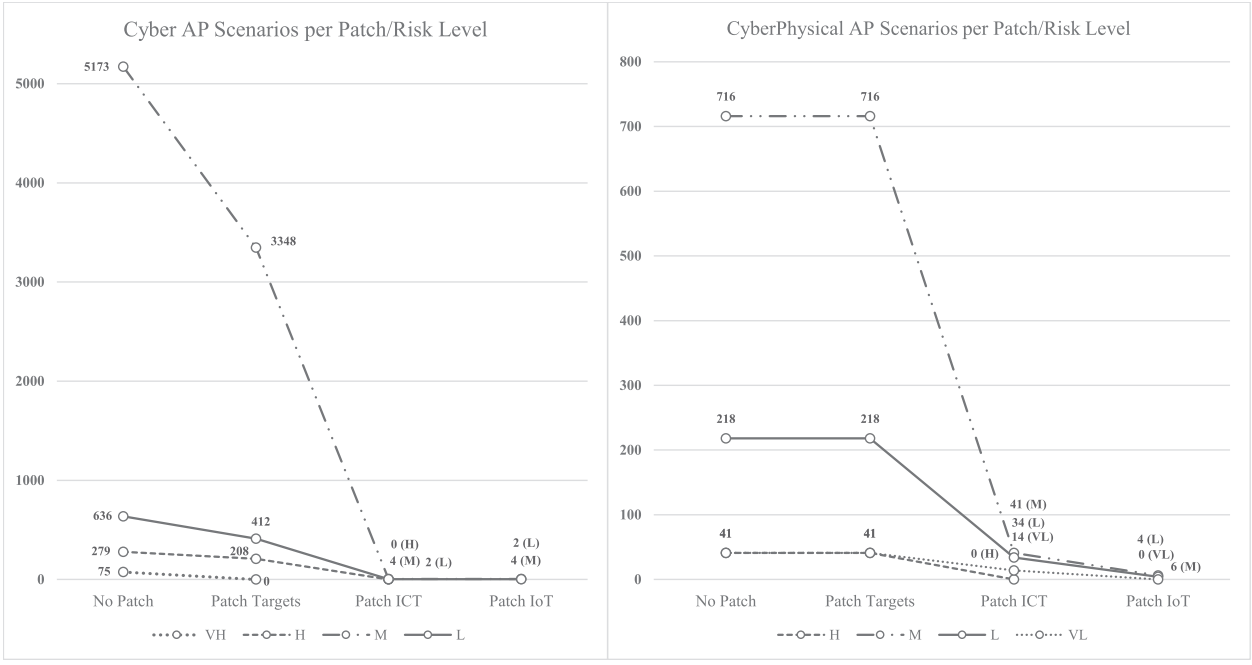

Fig. 8 – Risk level and multitude of attack path scenarios per patch level.

formation and other stable datasets. A promising approach for the production of stable datasets such as the CVSS temporal and environmental scores and the adversarial (threat agent) characteristics, is the utilization of Natural Language Processing (NLP) and other Machine Learning techniques to parse and create context from existing open sources. Finally, we plan to investigate the potential of integrating publicly available threat intelligence sources (e.g. Stergiopoulos et al. (2018)).

Declaration of Competing Interest

We declare no conflicts of Interest.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.cose.2021.102316

CRediT authorship contribution statement

Ioannis Stellios: Conceptualization, Methodology, Validation, Formal analysis, Investigation, Resources, Writing - original draft, Writing - review & editing. **Panayiotis Kotzanikolaou:** Conceptualization, Methodology, Formal analysis, Writing - original draft, Writing - review & editing, Supervision, Project administration, Funding acquisition. **Christos Grigoriadis:** Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing - review & editing, Visualization.

REFERENCES

Agadakos I, Chen C-Y, Campanelli M, Anantharaman P, Hasan M, Copos B, Lepoint T, Locasto M, Ciocarlie GF, Lindqvist U. Jumping the air gap: modeling cyber-physical attack paths in the internet-of-things. In: Proceedings of the 2017 Workshop on Cyber-Physical Systems Security and PrivaCy. ACM; 2017. p. 37–48.

Al Ghazo AT, Ibrahim M, Ren H, Kumar R. A2G2V: automatic attack graph generation and visualization and its applications to computer and scada networks. IEEE Trans. Syst. Man Cybern. 2019.

Alhanahnah M, Stevens C, Bagheri H. Scalable analysis of interaction threats in IoTsystems. In: Proceedings of the 29th ACM SIGSOFT International Symposium on Software Testing and Analysis; 2020. p. 272–85.

Ammann P, Wijesekera D, Kaushik S. Scalable, graph-based network vulnerability analysis. In: Proceedings of the 9th ACM Conference on Computer and Communications Security; 2002. p. 217–24.

Assange, J., 2017. Vault 7: cia hacking tools revealed. WikiLeaks.(Mar. 2017). Retrieved Mar 7, 2017.

Barker, W., NIST, S., 2008. 800-60, revision 1. Guide for mapping types of information and information systems to security categories.

Cheminod M, Bertolotti IC, Durante L, Maggi P, Pozza D, Sisto R, Valenzano A. Detecting chains of vulnerabilities in industrial networks. IEEE Trans. Ind. Inf. 2009;5(2):181–93.

Cheng P, Wang L, Jajodia S, Singhal A. Aggregating cvss base scores for semantics-rich network security metrics. In: 2012 IEEE 31st Symposium on Reliable Distributed Systems. IEEE; 2012. p. 31–40.

Dorsemaine B, Gaulier J-P, Wary J-P, Kheir N, Urien P. A new threat assessment method for integrating an IoTinfrastructure in an information system. In: Distributed Computing Systems Workshops (ICDCSW), 2017 IEEE 37th International Conference on. IEEE; 2017. p. 105–12.

Erdősi, P. M.,. The common vulnerability scoring system (CVSS) generations–usefulness and deficiencies.
Fiebig T, Krissler J, Hänsch R. In: 8th {USENIX} Workshop on Offensive Technologies ({WOOT} 14). Security impact of high resolution smartphone cameras; 2014.
FIRST.Org, 2019. Common vulnerability scoring system v3.1: user guide.
Force JT. In: Technical Report. Security and Privacy Controls for Information Systems and Organizations. National Institute of Standards and Technology; 2017.
Force JT, Initiative T. Security and privacy controls for federal information systems and organizations. NIST Spec. Publ. 2013;800(53):8–13.
Gallon L, Bascou JJ. Using CVSS in attack graphs. In: 2011 Sixth International Conference on Availability, Reliability and Security. IEEE; 2011. p. 59–66.
Ge M, Hong JB, Guttmann W, Kim DS. A framework for automating security analysis of the internet of things. J. Netw. Comput. Appl. 2017;83:12–27.
George G, Thampi SM. Vulnerability-based risk assessment and mitigation strategies for edge devices in the internet of things. Pervasive Mob. Comput. 2019;59:101068.
Gritzalis D, Iseppi G, Mylonas A, Stavrou V. Exiting the risk assessment maze: a meta-survey. ACM Comput. Surv. 2018;51(1):11.
Group, J. T. F. T. I. W., et al., 2012. Nist sp 800-30: guide for conducting risk assessments.
Guri M, Bykhovsky D. Air-jumper: covert air-gap exfiltration/infiltration via security cameras & infrared. Comput. Secur. 2019;82:15–29.
Ingols K, Chu M, Lippmann R, Webster S, Boyer S. Modeling modern network attacks and countermeasures using attack graphs. In: 2009 Annual Computer Security Applications Conference. IEEE; 2009. p. 117–26.
Jajodia S, Noel S, O'berry B. Topological analysis of network attack vulnerability. In: Managing Cyber Threats. Springer; 2005. p. 247–66.
Kott A, Ludwig J, Lange M. Assessing mission impact of cyberattacks: toward a model-driven paradigm. IEEE Secur. Privacy 2017(5):65–74.
Lallie HS, Debattista K, Bal J. A review of attack graph and attack tree visual syntax in cyber security. Comput. Sci. Rev. 2020;35:100219.
Li E, Kang C, Huang D, Hu M, Chang F, He L, Li X. Quantitative model of attacks on distribution automation systems based on CVSS and attack trees. Information 2019;10(8):251.
Liu C, Zhang Y, Zeng J, Peng L, Chen R. Research on dynamical security risk assessment for the internet of things inspired by immunology. In: Natural Computation (ICNC), 2012 Eighth International Conference on. IEEE; 2012. p. 874–8.
Maggi F, Quarta D, Pogliani M, Polino M, Zanchettin AM, Zanero S. In: Technical Report. Rogue Robots: Testing the Limits of an Industrial Robot's Security. Trend Micro, Politecnico di Milano; 2017.
for Network, E. U. A., (ENISA), I. S., 2020. Main incidents in the eu and worldwide – enisa threat landscape.
O'Flynn CP. Message denial and alteration on IEEE 802.15.4 low-power radio networks. In: New Technologies, Mobility and Security (NTMS), 2011 4th IFIP International Conference on. IEEE; 2011. p. 1–5.
Petit J, Stottelaar B, Feiri M, Kargl F. Remote attacks on automated vehicles sensors: experiments on camera and Lidar. Black Hat Europe 2015;11:2015.
Phillips C, Swiler LP. A graph-based system for network-vulnerability analysis. In: Proceedings of the 1998 Workshop on New Security Paradigms; 1998. p. 71–9.
Proofpoint, 2019. Protecting patients, providers and payers 2019 healthcare threat report.
ProofPoint, 2020. 2020 healthcare threat landscape.
Ronen E, O'Flynn C, Shamir A, Weingarten A-O. IoT goes nuclear: creating a zigbee chain reaction. IACR Cryptol. ePrint Arch. 2016;2016:1047.
Ronen E, Shamir A. Extended functionality attacks on IoT devices: the case of smart lights. In: 2016 IEEE European Symposium on Security and Privacy (EuroS&P). IEEE; 2016. p. 3–12.
Scroxton, A., 2020. German authorities probe ransomware hospital death.
SecurityScorecard, D., 2020. Listening to patient data security: Healthcare industry and telehealth cybersecurity risks.
Sequeiros JB, Chimuco FT, Samaila MG, Freire MM, Inácio PR. Attack and system modeling applied to IoT, cloud, and mobile ecosystems: embedding security by design. ACM Comput. Surv. 2020;53(2):1–32.
Sheyner O, Haines J, Jha S, Lippmann R, Wing JM. Automated generation and analysis of attack graphs. In: Proceedings 2002 IEEE Symposium on Security and Privacy. IEEE; 2002. p. 273–84.
Stellios I, Kotzanikolaou P, Psarakis M, Alcaraz C, Lopez J. A survey of IoT-enabled cyberattacks: assessing attack paths to critical infrastructures and services. IEEE Commun. Surv. Tutor. 2018;20(4):3453–95.
Stergiopoulos G, Gritzalis D, Kouktzoglou V. Using formal distributions for threat likelihood estimation in cloud-enabled it risk assessment. Comput. Netw. 2018;134:23–45.
Stouffer K, Falco J, Scarfone K. Guide to industrial control systems (ICS) security. NIST Spec. Publ. 2011;800(82). 16–16
Zambon E, Etalle S, Wieringa RJ, Hartel P. Model-based qualitative risk assessment for availability of it infrastructures. Softw. Syst. Model. 2011;10(4):553–80.

Ioannis Stellios received the Diploma degree in engineering and the M.B.A. degree from the National and Technical University of Athens, Greece, in 2002 and 2009, respectively, as well as the M.Sc. degree in ICT security from the Department of Informatics, University of Piraeus, in 2015, where he is currently pursuing the Ph.D. degree. His main research interests include IoT security and critical infrastructure protection.

Panayiotis Kotzanikolaou is an Associate Professor in Network Security and Privacy at the University of Piraeus, Department of Informatics. He has obtained a Degree in computer science (1998) from the University of Piraeus and a Ph.D in ICT security (2003). Formerly, has served as a Security Auditor at the Hellenic Authority for the Security and Privacy in Communications (ADAE), and has also worked as a security consultant in the private sector. He has participated in various national and European R&D projects. He has participated as a Program Committee member in international conferences and he is a reviewer in various international journals. He has published more than 70 papers in books, journals and international conferences. He is a member of the Greek Computing Society and has received various certifications in information security (CISSP, ISO 27001 Lead Auditor).

Christos Grigoriadis received his M.Sc. degree in ICT security from the Department of Informatics, University of Piraeus, in 2018, where he is currently pursuing the Ph.D. degree. His main research interests include the application threat modeling and network security.