Designing Secure and Resilient Cyber-Physical Systems: a Model-based Moving Target Defense Approach

Designing Secure and Resilient Cyber-Physical Systems: a Model-based Moving Target Defense Approach

Valentina Casola, Alessandra De Benedictis
Department of Electrical Engineering and Information Technology
University of Naples Federico II, Naples, Italy
Email: {casolav,alessandra.debenedictis}@unina.it
Carlo Mazzocca, Rebecca Montanari
Department of Computer Science and Engineering
Alma Mater Studiorum University of Bologna, Bologna, Italy
Email: {carlo.mazzocca,rebecca.montanari}@unibo.it

Abstract—Cyber-physical systems (CPSs) rely upon the deep integration of computation and physical processes/systems, enabled by Internet of Things (IoT), edge computing, and cloud technologies. Noticeably, cybersecurity is a major concern in CPSs, since attacks may exploit both cyber and physical vulnerabilities and damage significantly physical equipment, compromise operational safety, and impact negatively on product quality and performance. In this context, CPS design should take both security and resilience requirements into account, by identifying the needed measures not only to prevent but also to withstand, recover from, and adapt to adverse conditions and attacks. The approach proposed in this paper aims at improving the security and resilience of a CPS deployment through a model-based design methodology leveraging security-by-design principles and Moving Target Defense (MTD) techniques, consisting in continually shifting a system configuration to reduce the attack success probability and survive attacks. Our methodology, in particular, is meant to support the threat modeling process of a CPS and the identification, based on spotted threats and on the properties of involved assets and data, of the security controls to include within the design to mitigate existing threats and of the MTD techniques to integrate in order to increase resilience.

Index Terms—CPS threat modeling, CPS system modeling, resilient and secure CPS design, moving target defense

1 Introduction

Cyber-physical systems (CPSs) rely upon the deep integration of computation ad physical processes/systems. In a CPS, the physical layer senses information from the surrounding environment and sends them to the cyber layer, which comprises computing resources aiming at controlling and monitoring the physical world with feedback loops. Information processed at the cyber layer are then employed to reconfigure system parameters and/or make changes to physical processes.

Technology advancements as well as the increasing availability of sensors and actuators are paving the way to CPSs across many sectors such as smart manufacturing, intelligent transportation, personalized health care, emergency response, and electric power generation and delivery. Application domains can be very different from each other and typically involve a large number of heterogeneous components that complicate the modeling of a CPS with reasonable fidelity [1]. Nevertheless, some key elements are usually in common to all CPS architectures. Indeed, it is noteworthy that cloud, edge, and Internet of Things (IoT), which converge towards the so-called *cloud continuum* paradigm, are enabling technologies that make interconnectivity possible and provide intelligence to these systems. From an architecture and technology perspective, a CPS can be represented through a layer-based model where each asset is bonded to a layer (cloud, edge, or IoT) [2] according to its computation capabilities and technology. One of the major concerns of such architectures is the huge number of connections and interactions among the layers, which considerably broaden the attack surface.

Noticeably, cybersecurity is a major concern in CPSs, since attacks may exploit both cyber and physical vulnerabilities and damage significantly physical equipment, compromise operational safety, and impact negatively on product quality and performance. In this context, CPS design should take security into account from the beginning, by identifying the needed security measures based on existing threats and by enforcing them directly into the system architecture, according to security-by-design principles. Unfortunately, identifying the actual threats that affect a given deployment is a challenging task, as risks vary significantly by individual use case and application domain, and heavily depend on involved components and technologies.

Moreover, typical security concerns related to the loss of control experienced in cloud environments add up with the issues affecting the edge and IoT layers, characterized by highly heterogeneous hardware platforms, operating systems, and communication protocols, different computation,

storage, and communication capabilities, typically coarse-grained access control mechanisms, and a general lack of attack awareness due to the limited interfaces usually offered by devices.

Another important aspect to take into account is that, even when appropriate security measures are enforced in a system to thwart existing threats, attacks and breaches may still happen and should be properly handled. This urges the need for both security and resilience measures, especially in the critical contexts where CPS are adopted. According to NIST [3], resilience (or resiliency) is "the ability to anticipate, withstand, recover from, and adapt to adverse conditions, stresses, attacks, or compromises on systems that use or are enabled by cyber resources". As outlined in a recent (December 2021) NIST publication (SP160 Volume 2 Rev1 - Developing Cyber-Resilient Systems: A Systems Security Engineering Approach) [4], cyber-resilience measures may have different goals, such as preventing/avoiding the successful execution of an attack, minimizing the degradation/interruption of delivered services, limiting current and preventing future damages, restoring compromised functionalities, and/or restructuring/modifying systems or subsystems to reduce risks. Suggested approaches to implement resilience within a system include, among others, dynamic reconfiguration, dynamic resource allocation, obfuscation and deception, diversity, and temporal or contextual unpredictability, which belong to the set of so-called Moving Target Defense (MTD) techniques.

MTD [5] is a proactive defense approach consisting in continually shifting a system configuration to increase the uncertainty for the attacker and reduce the attack success probability. MTD strategies can help not only increase the security of a system but also provide it with resilience capabilities. They enable to *anticipate* attackers' moves by proactively shifting the system's attack surface to thwart reconnaissance attempts, and to *adapt* to adverse conditions, *preserve system availability* and *recover from attacks* by suitably reconfiguring the system at different levels, relying upon diversity and redundancy techniques.

The approach proposed in this paper aims at improving the security and resilience of a CPS deployment through a model-based design methodology leveraging security-by-design principles and MTD techniques. In particular, our contribution is two-fold:

- to improve CPS security, we *introduce a semi-automated threat modeling and countermeasure selection methodology* meant to help developers identify the security controls to include within the design in order to mitigate existing threats;
- to improve CPS resilience, we leverage MTD principles to *associate different MTD techniques/strategies with system assets* based on existing threats and on asset properties.

Our approach leverages a comprehensive system model able to describe the main involved architectural elements (assets) and the associated data flow, with a focus on the specific properties that may impact on the applicability of threats and of associated countermeasures and related MTD strategies. To the best of our knowledge, we are the first to propose a model-based moving target defense approach for CPS.

The reminder of this paper is structured as follows. Section 2 presents an overview of existing MTD techniques and approaches aimed to model and assess the security in CPS systems. Section 3 illustrates the system model behind our proposal, while Section 4 describes the proposed approach that comprises the identification of tailored threat, security controls identification, and MTD strategies. In Section 5, we apply our methodology to a CPS-based architecture. Finally, Section 7 draws our conclusions.

2 Background and Related work

In this section, we provide the needed background on Moving Target Defense and on the application of MTD techniques in cloud, edge, and IoT architectures. Moreover, we review some existing approaches to security analysis and design of CPSs, with particular reference to the IoT and edge layers.

2.1 Moving Target Defense

Moving Target Defense is a proactive cyber-defense paradigm according to which a system's attack surface is continually changed over time by means of reconfiguration, to increase complexity and cost for attackers during reconnaissance activities, limit the exposure of vulnerabilities, and increase overall system resilience. System reconfiguration can be applied at different levels and by leveraging different techniques, but existing approaches basically rely upon one or more of the following principles:

- *diversity*: different implementations of the system are used to deliver the same functionality. The different implementations are not affected by the same vulnerabilities and weaknesses, and it is difficult for an attacker to use the knowledge possibly gathered on a system implementation to damage also the other versions.
- *redundancy*: multiple replicas of the system (services, nodes, paths) are used to deliver the final service to customers. The way replicas are actually invoked/activated is unpredictable for the attacker, which cannot target a specific replica to attack. This technique is particularly suited against denial of service attacks.
- *shuffling*: system settings at various layers are rearranged during system operation (e.g., address randomization, instruction set randomization, live VM migration, topology rearrangement). This technique is maybe the most effective one, but it is also the most expensive and complex and it is not always viable in real systems.

Independently of the reconfiguration scope and of the actual reconfiguration techniques adopted to change the attack surface of a system, MTD approaches can be distinguished into two main classes depending on when reconfiguration is triggered: in fact, it may be either launched periodically on a time basis or triggered by specific events, for example upon detection of attack attempts. In general, MTD techniques may either affect a single processing node (e.g., changing the VM used to deploy a given software component) or impact on a larger portion of a system (e.g., changing the topology of a network or the address used by all the system nodes) and, by their nature, they are characterized by different costs (in terms of both implementation cost and operation

overhead) and benefits (in terms of entropy and difficulty in completing a successful attack).

Several MTD approaches have been proposed in the literature in the last decade. As pointed out in [6], existing solutions mostly fall into two main categories: on the one hand, we find techniques aimed at shifting the so-called *exploration surface*, by presenting the attacker with a noisy or inaccurate view of a system to thwart reconnaissance activities (e.g., using IP address randomization, network links obfuscation or by feeding possible attackers with false information about a system); on the other hand, there some techniques focus on shifting the actual *attack surface*, with the goal of invalidating attacks *after* reconnaissance (e.g., by dynamically switching among different Operating Systems, different application implementations, different binaries, different physical machines for Virtual Machine deployment, etc.).

The application of moving target defense strategies in the cloud environment has been addressed by several research studies in the last years. As pointed out in [7], most of the proposed MTD techniques leverage cloud computing inherent features. Often, proposed strategies entail a reconfiguration of the VMs used for the service deployment, and consider proper migration techniques to ensure the service correct operation. Other approaches leverage Software-defined networking [8], [9] or container technology [10], [11]. In [12], a Security SLA-driven MTD framework was presented, enabling to automatically switch among different admissible application configurations while preserving the application Security SLA. Admissible configurations are obtained by applying diversity, redundancy, and shuffling principles to the application *logic layer*, including the application components' implementation, the application architecture and the communication protocols, and to the application *deployment layer*, including virtual machines, virtualization services, and physical hardware.

Several MTD techniques have been proposed targeting IoT and edge devices. In [13], the authors review the existing recent literature on MTD techniques for IoT, showing that they are actually feasible even if generally harder to implement due to resource constraints. Based on the analysis made by the authors, most of the existing approaches entail reconfiguration at the network layer, in terms of protocols, addresses, or topology. Other popular approaches shift the runtime environment (reconfiguration of RAM addresses and instruction set), the software (e.g., the binary image of the application), the data format, or the platform (VM, Operating System). In [14] and [15], the authors proposed an approach for reconfiguring resource-constrained devices at two different levels, namely at the cryptosystem level (by switching among different cryptographic protocols or cryptographic keys) and at the firmware level (by shifting among different application binary image), by evaluating the impact of reconfiguration on performance and energy consumption constraints.

2.2 CPS Security analysis and modeling

As anticipated, CPSs consist of multiple layers and assets, which makes it challenging to identify and eliminate all possible vulnerabilities. In recent years, some research efforts have been spent to cope with the security analysis of complex CPSs. The authors in [16], for example, proposed a comprehensive threat modeling methodology that comprises an adversary model and an attack model. The proposed attack model extends MITRE taxonomy with some additional concepts, which do not take into account asset properties, typologies, and sensitivity of data. In [17], instead, the author presented a framework based on the STRIDE classification to perform the threat modeling of a CPS. The proposed methodology is scarcely automatable since threats and security countermeasures are identified by experts who have a deep knowledge of the system.

With specific regard to the IoT and edge layers, a few approaches have been proposed in the literature devoted to their security design and analysis. IoT Sentinel [18] is a system capable of automatically identifying the types of devices and subsequently establishing which rules should be enforced to constrain the communications of devices affected by potential security vulnerabilities. This security system allows minimizing the damage resulting from vulnerable devices. A framework for modeling and assessing the security of IoT was proposed in [19]. It is employed to build a graphical security model aiming at capturing potential attack paths in the network. Furthermore, the authors provide a security evaluator responsible for automating the security analysis. The results of the assessment of the security level of the IoT network provide a clearer picture of which assets and paths should be protected at first. Then, the defense strategies are compared to choose the most effective device-level security strategies. Soteria [20] is a static analysis system proposed to validate whether an IoT application or IoT environment adheres to identified security, safety, and functional properties. It translates platform-specific IoT source code into an intermediate representation and then extracts a state model on which verifies the desired properties. In [21], the authors propose an IoT Security Model (IoTSM) that allows organizations to plan and implement a strategy for developing end-to-end IoT security. This approach also enables analyzing, describing, and measuring the security posture, level, and practice of an IoT organization. Most of the current state-of-art research efforts target security issues, challenges, and frameworks for securing edge computing systems [22], whereas the area of automated threat modeling is still in its infancy. In this direction, the authors in [23] proposed a semi-automated threat modeling fine-grained approach for edge computing systems that enables identifying threats as well as the security controls to enforce. Our proposal represents an original approach since, to the best of our knowledge, we are the first to propose a semi-automated threat modeling approach for CSPs that explicitly takes into account the data flow and some relevant features of the components to determine threats as well as their security mitigation and alternative security techniques.

3 System Modeling

As mentioned in the Introduction, CPSs are hard to model due to the significant number of heterogeneous components and interconnections belonging to different architectural layers and leveraging different technologies. From a high

level point of view, a CPS can be represented as a set of interconnected layers that collaborate to achieve a common goal [24]. As illustrated in Figure 1, our system modeling proposal goes exactly in this direction since we model a CPS by means of a three-layer model namely, *cloud layer*, *edge layer*, and *device layer*. The cloud layer addresses heavy burden tasks such as big data analytics or data mining. On the other hand, the edge layer employs computing resources to control and monitor physical processes at the outskirt of the system, as close as possible where data are generated. This layer is particularly relevant for latency-sensitive applications where the delay resulting from data traveling back and forth between the devices and the cloud data centers may gave a negative impact on system functionality and/or on safety. Finally, the device layer refers to physical components that sense or modify the physical world.

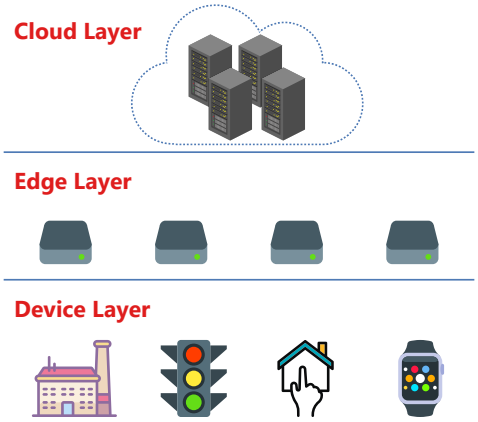


Fig. 1. CPS three-layer model.

This layered modeling can be applied to several CSP-based scenarios. For example, we discuss how it can be applied to Industry 4.0 where CPSs are now playing a key role. Historically, enterprise architectures have been designed to follow the Purdue Reference Model [25]. This model is particularly suited to outline the interconnections and interdependencies of all the main components of a typical Industrial Control System (ICS) since it splits the ICS architecture into two zones: IT and OT. The Purdue Model foresees the following levels:

- **Level 0 - Physical Process**: physical components (e.g., sensors and actuators) that build products;
- **Level 1 - Control**: systems that monitor and send commands to the devices at Level 0. This level includes Programmable Logic Controllers (PLCs), Remote Terminal Units (RTUs), and Intelligent Electronic Devices (IEDs);
- **Level 2 - Supervisory**: supervisor systems that control the overall processes. For example, human-machine interfaces (HMIs) and engineering workstations;
- **Level 3 - Manufacturing Operations Systems**: systems supporting the management of production workflows such as batch management and data historians;
- **Level 4/5 - Enterprise**: enterprise network, it collects data from ICS systems for business decisions.

The main strength of the Purdue model is the rigid hierarchy that makes it clear the role of each component within smart manufacturing by associating it to a level according to its functionality. On the other hand, modern control system networks are not always clearly separated from everything else. Furthermore, the increasing demands for real-time OT data and the integration of cloud/edge systems, and backend services are paving the way to more flexible systems. However, since this model can still be useful to understand how information flows and to identify potential attack vectors, some revised versions have been proposed. For example, the European Union Cyber Security Agency (ENISA) presented a Purdue Model that recognizes a Level 3 Industrial IoT (IIoT) platform that communicates directly with Level 1 IIoT devices.

The three-layer modeling adopted in this paper addresses different needs, including data management and security, and simplifies the description of a modern industrial system. Nevertheless, the levels of the Purdue model can be easily mapped onto our three-layer model. In particular, Level 0 which mainly comprises sensors and actuators corresponds to the device layer. The levels ranging from 1 to 3, which comprises elements physically deployed within the manufacturing implant and aim at managing and controlling devices and/or communication with the IT network, are associated with the edge layer. Finally, the IT zone is bound to the cloud layer.

Hence, we formally model a modern CPS as a set $\langle \mathcal{A}, \mathcal{D} \rangle$ where $\mathcal{A}$ denotes the set of assets and $\mathcal{D}$ is the set of data stored and/or transmitted; they will be described in the following subsections.

3.1 Asset Characterization

As sketched in the model in Figure 2, CPS architectures comprise several types of assets belonging to different architectural and functional levels. In particular, the set of assets $\mathcal{A}$ is composed of the following finite, disjoint, non-empty sets:

- the set of physical/virtual processing nodes $\mathcal{N}$;
- the set of software components/modules $\mathcal{S}$;
- and the set of communication channels $\mathcal{C}$.

By adopting this notation, the set of assets can be formalized as: $\mathcal{A} = \mathcal{N} \cup \mathcal{S} \cup \mathcal{C}$. Assets are characterized by a set of *properties* that depend on their type. Let $\mathcal{P}$ be the set of all possible asset properties, which comprises the following finite, disjoint, non-empty subsets:

- the set of physical/virtual node properties $\mathcal{P}_\mathcal{N}$, which can assume the values in $\mathcal{V}_\mathcal{N}$;
- the set of software component/module properties $\mathcal{P}_\mathcal{S}$, which can assume the values in $\mathcal{V}_\mathcal{S}$;
- and the set of communication channel properties $\mathcal{P}_\mathcal{C}$, which can assume the values in $\mathcal{V}_\mathcal{C}$.

Thus, the set of asset properties is $\mathcal{P} = \mathcal{P}_\mathcal{N} \cup \mathcal{P}_\mathcal{S} \cup \mathcal{P}_\mathcal{C}$, while we denote with $\mathcal{V} = \mathcal{V}_\mathcal{N} \cup \mathcal{V}_\mathcal{Q} \cup \mathcal{V}_\mathcal{C}$ the set of all admissible property values. In the following, we discuss assets as well as their properties.

3.1.1 Physical/Virtual Processing Nodes

The first asset category comprises the set $\mathcal{N}$ of physical and virtual processing nodes belonging to different layers of the proposed model. These nodes are both devoted to running

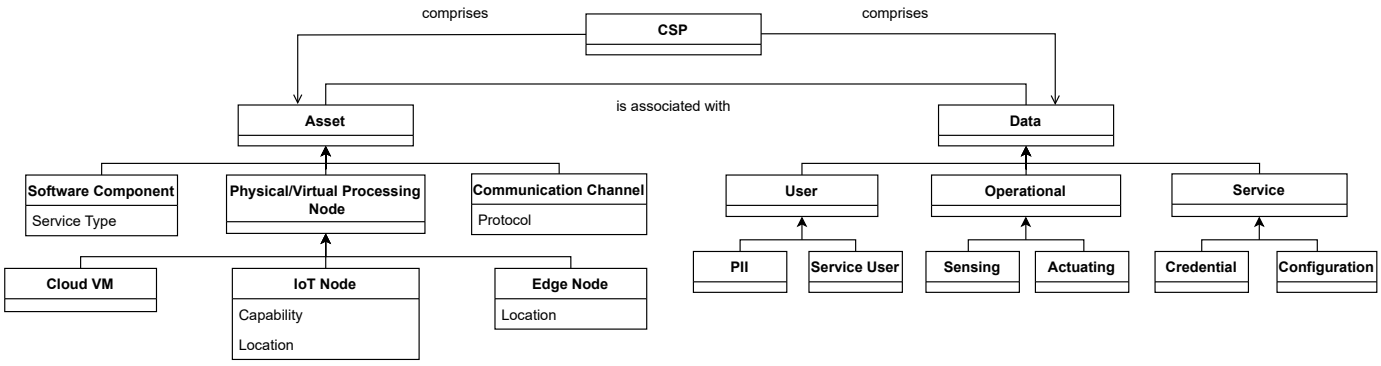


Fig. 2. Asset and data modeling.

supervisory and control software, as well as enterprise applications (e.g., analyze data and make business decisions). At the device layer, they are represented by physical devices such as sensors and actuators. Typically, processing nodes that belong to the edge layer are represented by hardware machines, while at the cloud layer they correspond to virtual machines offered according to the Infrastructure-as-a-Service (IaaS) paradigm. Therefore, the set of processing nodes $\mathcal{N}$ considered in this paper can be specified as: $\mathcal{N}$ = {CLOUD VM, EDGE NODE, IOT NODE}.

It is worth reminding that processing nodes offer different storage and computational capabilities, that enable the implementation of different security mechanisms. This is a key aspect that must be taken into account during the countermeasure selection process since these features enable the identification of the set of feasible security controls on a specific asset. Cloud-based computing resources and edge nodes provide sufficient capabilities to support traditional security mechanisms, while IoT devices are usually characterized by limited computing and storage capabilities that often only allow for the implementation of simple and lightweight protocols and mechanisms. On the other hand, applicable threats often depend on some physical characteristics of nodes: for instance, having a battery-powered device opens up to specific threats that do not apply to AC-powered nodes (e.g., battery exhaustion).

Based on the above considerations, IoT devices can be *labeled* according to their computing/storage capabilities and power supply by means of a `capability` property. In particular, following the classification proposed in [26], devices can be distinguished in *Constrained*, *Limited*, *Restricted*, and *Normal*. Constrained devices (battery-powered, up to 10KB RAM, and up to 128KB ROM) are the weakest and do not support any security mechanisms. Limited devices (battery-powered, 10-32KB RAM, and 128-512KB ROM) can support some symmetric key-based protocol. Restricted devices (battery/AC powered, 32-128KB RAM, and 512KB-10MB ROM) are more powerful devices able to implement symmetric protocols and lightweight asymmetric key-based protocols. Finally, normal devices (AC-powered, 128KB and above RAM, 10MB ROM and above) are powerful devices able to implement any traditional security protocols. Besides computing/storage capabilities and power supply characteristics, in some scenarios also the location where a node is physically deployed impacts on its security, as it may lead to specific threats. In order to take this aspect into account, it is possible to identify another node property named `location`, which can assume two values: *Protected* and *Open*. The former refers to assets that are placed in an area that can be only accessed by authorized personnel, while the latter is related to assets that can be accessed without any restriction, and that therefore are more exposed to potential attackers. We denote the set of properties that can be associated with physical/virtual nodes as $\mathcal{P}_{\mathcal{N}}$ = {LOCATION, CAPABILITY} and the set of values that can be assumed by such properties as $\mathcal{V}_{\mathcal{N}}$ = {OPEN, PROTECTED, CONSTRAINED, LIMITED, RESTRICTED, NORMAL}.

3.1.2 Communication Channel

The second asset category includes the communication channels $\mathcal{C}$ established among nodes $\mathcal{N}$, hence, $\mathcal{C} \subseteq \mathcal{N} \times \mathcal{N}$. Since CPS are employed in many different scenarios, there are many communication protocols (ZigBee, MODBUS, Wi-Fi, etc) across layers that can be exploited by malicious attackers to compromise the system. To take into account specific threats associated with the communication channel, we assigned the `protocol` property to this asset, which corresponds to the protocol used for communication. Therefore, the set of communication channel properties $\mathcal{P}_{\mathcal{C}}$ only comprises {PROTOCOL}, while the set of values, which a protocol can assume, is denoted by $\mathcal{V}_{\mathcal{C}}$ = {ZIGBEE, MODBUS, WIFI, BLE}.

3.1.3 Software Component

The third asset category comprises the software components, modules, and services $\mathcal{S}$. It is clear that due to the increasing digitalization, the software is now playing a crucial role since it is at every level of a CPS. Software components can be very heterogeneous not only in terms of technology and complexity but also for their application domain.

In order to simplify the characterization of software components, we identified a `service type` property that can assume one of three possible values, namely: *web-based service*, if the component is primarily devoted to processing and exposes its services through a web (HTTP) interface (it is the case of cloud-based services and of web interfaces exposed by edge nodes to communicate with the cloud layer), *storage service*, if the component stores structured

or unstructured information (e.g., an on-premise DBMS, a cloud-based storage service, a key-value store, etc.), and *IoT service*, in case of services/applications running on IoT or edge nodes and accessible via non-HTTP protocols. This category also includes software employed to monitor and supervise the physical processes. The set of software component properties is denoted by $\mathcal{P}_{\mathcal{S}}$ = {SERVICE TYPE}, while the set of values that a service type can assume is denoted by $\mathcal{V}_{\mathcal{S}}$ = {WEB-BASED SERVICE, STORAGE SERVICE, IOT SERVICE}.

3.2 Data Characterization

Nowadays, data are extremely valuable information that have to be properly managed. To provide effective modeling of the data that flow in a CPS, we introduce a data classification based on the entities to which data are related. In particular, the set $\mathcal{D}$ of the data stored, processed and transmitted by a CPS comprises the following finite, disjoint, non-empty subsets (refer to Figure 2):

- the set of data related to users $\mathcal{U}$;
- the set of data related to operating environments $\mathcal{O}$;
- the set of data used for service operation $\mathcal{Q}$.

We denote the set of data by $\mathcal{D} = \mathcal{U} \cup \mathcal{O} \cup \mathcal{Q}$. Before analyzing in the following subsections each data category, let us outline that, independently of their source and of who/what they are concerned, data should be also classified according to their sensitivity. As stated by the ISO27001 standard, each organization should contemplate an information classification process to assess the data managed and the level of protection deemed. To address this need, we introduced a `sensitivity` property, to be assigned to any type of data. The possible values of the data sensitivity property are defined as: $\mathcal{P}_{\mathcal{D}}$ = {PUBLIC, INTERNAL, CONFIDENTIAL}. Public data (e.g., temperatures of public places, traffic condition data, etc.) can be accessed by everyone, internal data (e.g., user profile data, configuration settings, etc.) are only available to certain entities of the system and, finally, confidential data (e.g., credentials, biometrics, financial data, etc.) can be only handled by the owner. Based on the sensitivity level, the unauthorized disclosure, alteration, or destruction of data would result in a low, moderate, or significant level of risk, respectively. Let us now illustrate the considered data classification.

3.2.1 User-related data

User-related data are information belonging to end-users. They include both information used to interact with the system (credentials, profiling information, etc.), referred to as *Service user data*, and personally identifiable information (*PII*), which enable to uniquely identify an individual. Hence, data belonging to users are defined as: $\mathcal{U}$ = {PII, SERVICE USER}. Since 2016, the European General Data Protection Regulation (GDPR) [27] established a set of rules aiming at defining what must be protected to preserve the privacy of individuals, so companies have been forced to keep PII information in safely and securely manner. Article 9[1] of the European GDPR outlines the personal data that cannot be processed without the explicit consent of the interested party. All the information revealing racial or ethnic origin, political opinions, religious or philosophical beliefs, or trade union membership belong to personal data. Furthermore, this category also includes other information such as biometric data aiming at uniquely identifying a person, data concerning health or sexual orientation, judicial information, electronic communication (IP addresses), geographic location, etc.

1. https://www.privacy-regulation.eu/en/article-9-processing-of-special-categories-of-personal-data-GDPR.htm

3.2.2 Operational data

In a CPS there are many information related to the physical environment where the system is deployed. Operational data may be either sensed by sensors or used to control actuators. Hence, we classify operation data into two types, namely *Sensing* and *Actuating*. Data related to physical environments are therefore $\mathcal{O}$ = {SENSING, ACTUATING}. These data may have different sensitivity features that depend on the specific application domain.

3.2.3 Service data

Finally, service data include any other data not directly related to end-users or environments, such as service credentials/tokens for protected service interactions (referred to as *Credential* service data) or configuration parameters (e.g., deployment information) used by services and applications, that we denote as *Configuration* service data. Therefore, service data can be formalized as $\mathcal{Q}$ = {CONFIGURATION, CREDENTIAL}.

4 Approach

The approach proposed in this paper aims at improving the security and resilience of a CPS deployment through a model-based design methodology leveraging security-by-design principles and moving target defense. In particular, it enables to carry out a guided threat modeling process of the system under study, which enables spotting existing security issues according to types and features of the assets and concerning information. This method also allows determining security countermeasures to implement to prevent or mitigate identified threats. Furthermore, since attackers may still be able to defeat implemented cybersecurity safeguards, our methodology helps to identify appropriate moving target defense techniques to improve the overall system resilience.

Our approach leverages both the *system model*, built by identifying the involved assets along with related properties and the information flow, as described in Section 3, and a reference *security data model*, sketched in Figure 3, which suitably links together the concepts related to assets, asset properties, threats, security controls, and MTD techniques.

4.1 Threat Modeling and Countermeasure Selection

As outlined by the model in Figure 3, to facilitate the threat modeling process, threats are directly associated with assets and filtered based on their properties and on involved data flow, which actually impact on their applicability. We denote with $\mathcal{T}$ the set of threats and with $\mathcal{SC}$ the set of security controls. We define a function F that models the

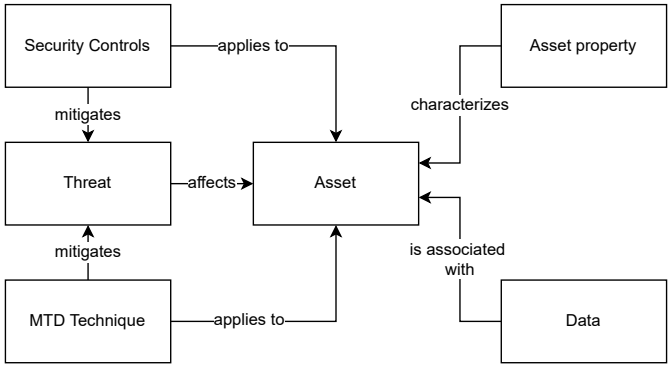


Fig. 3. Security data model.

applicability of threats to asset. Given an asset $a \in \mathcal{A}$, the set of related properties $P \in \mathcal{P}$, and of the data $D \in \mathcal{D}$ stored/transmitted, $F(a, P, D, \mathcal{T})$ returns the set of applicable threats $\mathcal{T}_a \subset \mathcal{T}$. Each threat $t \in \mathcal{T}_a$ if t is associated with a, P, and D.

Countermeasure selection consists in identifying the set of technical security controls to apply to an asset in order to mitigate existing threats. This step is accomplished thanks to the association between threats, assets, and standard security controls. Security controls are implemented through specific security mechanisms, which may negatively impact on the system performance or may be simply not viable due to resource constraints. To take into account this issue, asset properties are used also to refine the set of applicable security controls based on capabilities (in terms of computational power and storage capacity) of the involved components. For example, a *constrained* IoT node, due to its computing and storage capabilities, will typically not be able to prove the authenticity of a message through public-key algorithms. On the other hand, a *normal* IoT node has satisfying resources to enforce such countermeasure. This information is made explicit for processing nodes by specifying the `capability` property, while for software components it depends on the capability specified for the processing node used for their execution. Hence, we define a function M that models the feasibility of security controls to mitigate threats. Given the same input parameters of F, except from $\mathcal{T}_a$ that is provided instead of $\mathcal{T}$, $M(a, P, D, \mathcal{T}_a)$ returns the set of feasible security controls $SC_{\mathcal{T}_a}$ able to mitigate threats in $\mathcal{T}_a$. Each security control $sc \in SC_{\mathcal{T}_a}$ if sc is associated with a and P.

4.2 MTD Techniques Identification

With regard to resilience requirements, as anticipated, our methodology allows identifying a set of MTD techniques that can be adopted to further avoid/mitigate a threat realization, even when suitable countermeasures have been deployed (i.e., those identified following the threat modeling process). As anticipated, MTD techniques may leverage spatial and temporal diversity, redundancy, and shuffling to dynamically change the system attack surface over time. From an architectural point of view, MTD techniques can be classified based on the level at which they work [13]: *platform-level* MTD techniques referring to computing platforms and execution environment levels (including the CPU architecture, the instruction set, the memory system, the Operating System), *software-level* MTD techniques concerning application logic, architecture or code and, finally, *network-level* MTD techniques associated with communication channels, including communication and security protocols and network addresses and topologies. Although MTD techniques are directly associated with assets and threats in our security data model, the actual applicability of an MTD strategy to a given asset depends on available capabilities. Therefore, MTD techniques are filtered according to threats, assets, and asset properties. For example, let us consider a *firmware reconfiguration* technique consisting in changing the entire application running on an IoT device, thus making it harder for an attacker to succeed in gaining complete control of the node. Such a technique requires that the node has a memory big enough for storing locally a second version of the firmware. Therefore, referring to our modeling, this strategy may be feasible for a normal IoT device, but it can not be adopted for those classified as constrained and restricted.

We denote with $\mathcal{M}$ the set of MTD techniques and define a function B that models whether a technique can be employed to improve the resilience of an asset or not. Given an asset $a \in \mathcal{A}$, the set of its properties $P \in \mathcal{P}$, and the set of applicable threats $\mathcal{T}_a$ returned by F, $B(a, P, \mathcal{T}_a)$ returns the set of applicable MTD techniques $\mathcal{M}_a \subset \mathcal{M}$. Each MTD technique $m \in \mathcal{M}_a$ if m is associated with a, P, and $t \in \mathcal{T}_a$.

4.3 The Threat Catalogue

The security data model is implemented through a complex knowledge base referred to as the *Threat Catalogue*, which at state comprises more than 150 different threats specific to cloud services, web-based applications, storage services, IoT devices, edge nodes, and network protocols, derived from existing standards and scientific studies such as [22], [28], [29], [30], [31], [32], [33]. Concerning security controls, the catalogue currently includes security controls belonging to the NIST Security Control Framework [34], which represent the measures to be applied to each asset to mitigate existing threats. The NIST framework contains over 900 unique security controls that encompass 18 control families, including both base controls and *control enhancements*, which strengthen the fundamental security capability of a base control. Some of such security control families refer to organizational controls, which are not of interest for this study and, therefore, have not been included in the countermeasure selection process. Finally, the catalogue includes a set of MTD techniques recently proposed in the literature [7], [9], [12], [13], [15] that may be applied at different architectural levels in order to design more resilient CPS-bases architectures. The current catalogue implementation[2] relies upon the well-known Microsoft Threat Modeling Tool[3] that was customized with the assets described in Section 3, as well as the related threats, countermeasure, security controls and MTD techniques. Tables 1 and 2 report an extract of the Threat Catalogue. The former refers

2. https://github.com/ci-ma/ThreatModeling-MTD-CPS
3. https://docs.microsoft.com/en-us/azure/security/develop/threat-modeling-tool

TABLE 1
Extract of the Threat Catalogue that outlines the relationship between assets, asset properties, threats, data, and NIST security controls.

#	Asset	Threat	STRIDE	Properties	Data	Data Sensitivity	NIST Security Controls
1	Edge Node	Camouflage	Spoofing	Location:*Open*	-	-	IA-3, IA-3(1, 3), ...
2	Edge Node	Node Replication	Spoofing	Location:*Open*	-	-	IA-3, IA-3(1,3), ...
3	IoT Node	Battery Draining	Denial of Service	Location:*Open*, Capability:*Constrained, Limited, Restricted*	-	-	PE-2, PE-3, ...
4	IoT Node	Exhaustion of Power	Denial of Service, Spoofing	Capability:*Constrained, Limited, Restricted*	-	-	PE-11
5	Communication Channel	Network Key Sniffing	Information Disclosure	Protocol:*Zigbee*	Credential	Confidential	SC-8, SC-13, ...
6	Communication Channel	Message Elimination	Information Disclosure, Spoofing, Tempering	-	-	Internal, Confidential	AC-17, SA-18, ...
7	Cloud VM	Denial of Service	Denial of Service	-	-	-	SC-5, ...
8	Software Component	Reverse Engineering	Information Disclosure	ServiceType:*web-based, IoT*	-	Internal	SR-9, SR-9(1), ...

TABLE 2
Extract of the Threat Catalogue that outlines the relationship between assets, asset properties, threats, and MTD techniques.

#	Asset	Threat	Properties	MTD Level	MTD Technique
1	Edge Node	Camouflage	-	Network	Identity Virtualization
2	Edge Node	Node Replication	-	Network	Identity Virtualization
3	IoT Node	Battery Draining	Capability:*Constrained, Limited, Restricted*	Network	Honeypot
4	IoT Node	Exhaustion of Power	Capability:*Constrained, Limited, Restricted*	Software	Crypto-protocol reconfiguration
5	Communication Channel	Network Key Sniffing	Protocol:Zigbee	Network	Dynamic Re-keying
6	Communication Channel	Message Elimination	-	Network	Dynamic Re-keying
7	Cloud VM	Denial of Service	-	Platform	VM Migration
8	Software Component	Reverse Engineering	ServiceType:*web-based, IoT*	Software	Code Obfuscation

to threats and security controls, while the latter to MTD strategies. An asset property that does not influence a threat and/or a MTD technique is simply omitted, while if a threat and/or a MTD technique do not depend on a field of the catalogue, that field is filled with "-". In Table 1, the last column reports some of the NIST security controls and control enhancements (reported in round brackets) that should be enforced to thwart or mitigate the corresponding threat. As mentioned above, security controls will be selected according to capabilities of the assets belonging to the deployment. The effectiveness of our approach relies upon the completeness of the catalogue, however, it can be easily extended to include new threats and cope with new issues.

5 Case Study

In this section, to provide a concrete example of the proposed approach, we discuss how it can be applied to a CPS case study represented by a smart home security monitoring system. The deployment under study was introduced in [35] and sketched in Figure 4. The system comprises several IoT devices belonging to the *perception layer*, employed to sense information from the surrounding environment. Such devices are connected to the control *control layer*, represented by a gateway component responsible for managing them. The gateway interacts with the *decision layer* that leverages a set of cloud-based services to analyze the data provided by the physical world to make intelligent decisions and optimize the whole system. Intuitively, the layers of the architecture under study can be easily mapped onto those of our three layer model described in Section 3. The association between the perception and the device layer and between the decision and the cloud layer is trivial. Moreover, the gateway of the closed-loop system that accepts control instructions from the decision layer can be mapped to the edge layer. As graphically sketched in the diagram in Figure 5, we adopted the system modeling approach discussed in Section 4 to model our case study application. In particular,

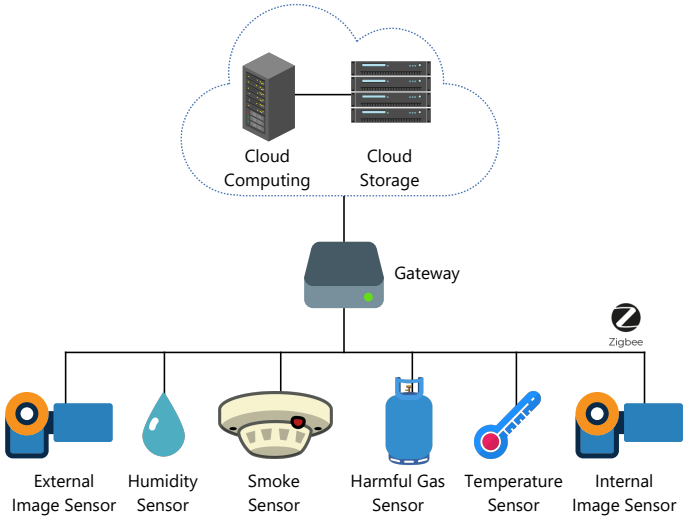


Fig. 4. Smart home security monitoring system CPS-based case study.

we identified and classified involved *assets* (depicted as rectangular solid boxes with different colors depending on their sub-type - yellow for IoT Nodes, light blue for Edge Nodes, light grey for Communication Channels, and green for Software Components) and specified related properties (reported as name-value attributes within each box). We also identified and classified involved *data* and associated them with relevant assets by also reporting the related sensitivity level (data are represented as dashed boxes linked to assets).

As shown in the diagram, the gateway node was modeled as an asset belonging to the Edge Node type, and its `location` property was set to *Protected* since the device is deployed inside the habitation and only the house owners can physically access it. With regard to the device layer, the image sensors, smoke sensor, temperature sensor, humidity sensor, and harmful gas sensor composing the home security monitoring system were modeled as assets belonging to the IoT Node type. Finally, the cloud-based web application and the underlying database service were modeled as assets of the Software Component type. Moreover, we modeled all communication channels between the device and edge layers as Communication Channel node types and specified the ZigBee `protocol` as a property of the node since it is the most commonly used protocol due to its network coverage, real-time data transmission, and cost characteristics. We specified HTTPS as the `protocol` used for communications between the gateway node and the upper layer and between the two cloud-based services. A more schematic view of part of the considered assets, data, and properties is provided in Table 3.

Since threat modeling and moving target defense approaches have been widely investigated in cloud and software design, in this work, we will focus on edge nodes and IoT devices, without carrying out any further analysis for other component types. For each asset a belonging to the home security system monitoring $\langle \mathcal{A}, \mathcal{D} \rangle$ we have to determine its properties $P \in \mathcal{P}$, as well as the typology and sensitivity of data $D \in \mathcal{D}$ stored and/or transmitted.

Once the CPS-based architecture has been modeled, we proceed with the threat identification and countermeasure selection. Moreover, moving target defense strategies are also retrieved. As mentioned above, such information are automatically collected through the knowledge base built by codifying security expertise. Each threat t applies to an asset a depending on its type, properties P, and concerned data D. Security controls and MTD techniques to mitigate threats are identified according to assets and their properties.

For example, with reference to the extract of the catalogue shown in Table 1, the gateway cannot be threatened by the *Camouflage* or *Node Replication* threats that affect nodes whose `location` is *Open*. However, it may be vulnerable to *Outage*, a threat that refers to edge nodes that stop performing their normal operation as they have been exposed to unauthorized access. With regard to involved data, due to its centralized nature, the gateway will handle all types of data, therefore the symbol "-" is used. All IoT devices, except for image sensors, are subject to *Battery Draining* and *Exhaustion of Power* threats. Independently of the adopted protocol, all communication channels, which transmit *Internal* and/or *Confidential* data, are always subject to *Message Elimination*. Furthermore, in the deployment under study, devices and the gateway interact using ZigBee as the communication protocol. Therefore, all network communications that involve *Credential* data, whose sensitivity is *Confidential*, are vulnerable to *Network Key Sniffing*. After having collected all threats, which threaten the CPS-based architecture under analysis, who is in charge of designing a secure and resilient system will have to ensure that corresponding security controls and MTD techniques will be properly implemented. For instance, the *Exhaustion of Power* threat affects the smoke sensor. In order to mitigate this threat, the *P-11 EMERGENCY POWER* security controls should be enforced. This control consists in providing "*an uninterruptible power supply to facilitate an orderly shutdown of the system and/or transition of the system to long-term alternate power in the event of a primary power source loss*". On the other hand, to improve the resilience of such a device while taking into account its energy limitations, the suggested MTD technique is the *Crypto-protocol Reconfiguration*, which is the least expensive in terms of energy consumption if compared to other strategies.

6 Discussion

The approach presented in this paper enables to support the threat analysis and countermeasure selection process for a generic CPS system based on a suitable system model and on an underlying knowledge base - the Threat Catalogue. The system model construction does not require specific security skills, as only general information on asset types and involved data must be specified, while security experts' knowledge is codified by the catalogue. This is particularly convenient in the case of limited economic resources, when costly security teams cannot be involved due to budget constraints but is generally desirable to drastically reduce the time needed for security-related activities.

However, it should be noted that the effectiveness of the proposed approach inherently depends on the *quality* of the Threat Catalogue, which codifies security experts' knowledge. In this regard, it is important to observe that full coverage of all possible threats and assets is not possible,

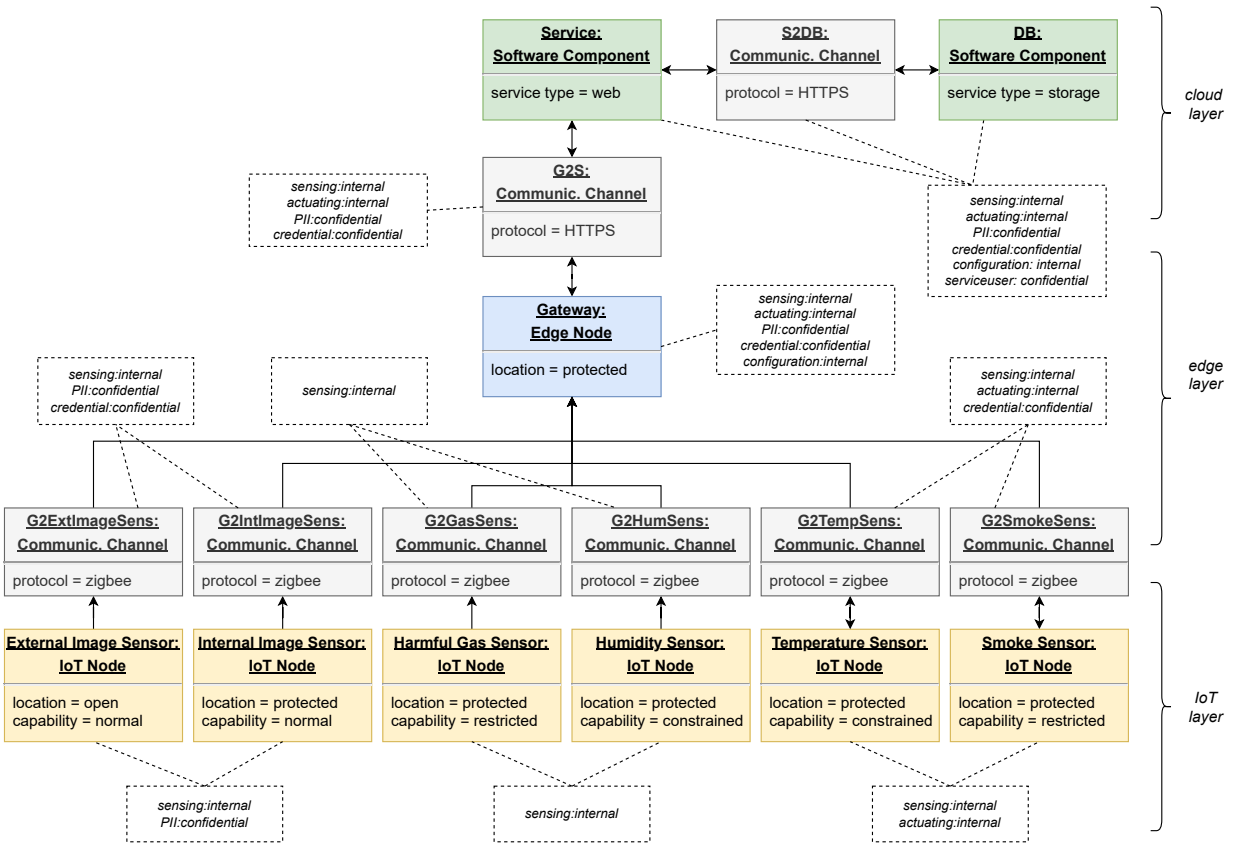


Fig. 5. Smart home security monitoring system CPS-based case study - system modeling

TABLE 3
An extract of the asset Specifications.

Asset	Asset Properties	Data Typology	Data Sensitivity
Gateway	Location:*Protected*	-	-
Humidity Sensor	Location:*Protected*, Capability: *Constrained*	Sensing	Internal
Internal Image Sensor	Location:*Protected*, Capability:*Normal*	Sensing, PII	Internal, Confidential
External Image Sensor	Location:*Open*, Capability:*Normal*	Sensing, PII	Internal, Confidential
Smoke Sensor	Location:*Protected*, Capability: *Restricted*	Sensing, Actuating	Internal
Temperature Sensor	Location:*Protected*, Capability: *Constrained*	Sensing, Actuating	Internal
Harmful Gas Sensor	Location:*Protected*, Capability: *Restricted*	Sensing	Internal
G2TempSens	Protocol:*ZigBee*	Credential, Sensing, Actuating	Internal, Confidential

due to the inherent dynamicity in the security threat and technological landscape. However, the catalogue is simply an instance of the security data model, which captures the most relevant concepts impacting on security risks. As such, it may be easily extended to take into account new assets, threats, security mechanisms, MTD techniques, and to better characterize assets and data with further properties. Apart from the catalogue, the underlying security data model may also be extended. For example, in a further step toward automation, selected measures may be enforced by means of suitable mechanisms seamlessly configured and activated in an as-a-service fashion based on available information contained in the catalogue. Therefore, we believe that the general approach validity is not dependent on the catalogue.

7 Conclusions and Future Work

Taking security into account from the very beginning of the development process is fundamental to obtain secure systems. It is mandatory in complex CPS scenarios, characterized by the integration of computation and physical processes/systems, where attacks may damage significantly physical equipment, compromise operational safety, and impact negatively on product quality and performance.

In this paper, we tackled the problem of designing secure and resilient CPSs and introduced a methodology aimed to support and facilitate the threat modeling process based on a suitable system description, able to take into account the features and properties of involved assets and data. Based on the system model and on an underlying security data model implemented by means of a complex knowledge base, our methodology enables designers to easily identify the measures to implement to improve both security and resilience. The latter, in particular, is ensured by selecting suitable moving target defense techniques based on available capabilities.

In our future work, we plan to further refine the system model to take into account more asset and data properties that impact on the security risk level and on the overall defense capabilities. Moreover, we plan to refine the countermeasure selection step by considering concrete security mechanisms and more specific actions tailored to the different assets instead of generic, technology-agnostic security controls. Finally, we aim at integrating more advanced strategies for the identification of the most suited MTD techniques to apply, based for example of additional application requirements related to latency and performance in general.

ACKNOWLEDGMENTS

This work has been partially funded from the University of Naples Federico II (Finanziamento delle Ricerca di Ateneo 2020).

REFERENCES

[1] P. Derler, E. A. Lee, and A. Sangiovanni Vincentelli, "Modeling cyber–physical systems," *Proceedings of the IEEE*, vol. 100, no. 1, pp. 13–28, 2012.

[2] K. Cao, S. Hu, Y. Shi, A. W. Colombo, S. Karnouskos, and X. Li, "A survey on edge and edge-cloud computing assisted cyber-physical systems," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 11, pp. 7806–7819, 2021.

[3] NIST, "Glossary - Cyber resiliency," 2018. [Online]. Available: https://csrc.nist.gov/glossary/term/cyber_resiliency

[4] NIST, "NIST Special Publication 800-160, Volume 2, Revision 1. Developing Cyber-Resilient Systems: A Systems Security Engineering Approach," 2021. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-160v2r1.pdf

[5] S. Jajodia, A. K. Ghosh, V. Swarup, C. Wang, and X. S. Wang, *Moving Target Defense: Creating Asymmetric Uncertainty for Cyber Threats*, 1st ed. Springer Publishing Company, Incorporated, 2011.

[6] S. Sengupta, A. Chowdhary, A. Sabur, A. Alshamrani, D. Huang, and S. Kambhampati, "A survey of moving target defenses for network security," *IEEE Communications Surveys Tutorials*, vol. 22, no. 3, pp. 1909–1941, 2020.

[7] M. Torquato and M. Vieira, "Moving target defense in cloud computing: A systematic mapping study," *Computers & Security*, vol. 92, p. 101742, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404820300286

[8] M. Villarreal-Vasquez, B. Bhargava, P. Angin, N. Ahmed, D. Goodwin, K. Brin, and J. Kobes, "An MTD-Based Self-Adaptive Resilience Approach for Cloud Systems," in *2017 IEEE 10th International Conference on Cloud Computing (CLOUD)*, 2017, pp. 723–726.

[9] A. Chowdhary, S. Pisharody, and D. Huang, "SDN Based Scalable MTD Solution in Cloud Network," in *Proceedings of the 2016 ACM Workshop on Moving Target Defense*, ser. MTD '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 27–36. [Online]. Available: https://doi.org/10.1145/2995272.2995274

[10] H. Jin, Z. Li, D. Zou, and B. Yuan, "DSEOM: A Framework for Dynamic Security Evaluation and Optimization of MTD in Container-Based Cloud," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 3, pp. 1125–1136, 2021.

[11] K. A. Torkura, M. I. Sukmana, A. V. Kayem, F. Cheng, and C. Meinel, "A cyber risk based moving target defense mechanism for microservice architectures," in *2018 IEEE Intl Conf on Parallel Distributed Processing with Applications, Ubiquitous Computing Communications, Big Data Cloud Computing, Social Computing Networking, Sustainable Computing Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom)*, 2018, pp. 932–939.

[12] V. Casola, A. De Benedictis, M. Rak, and U. Villano, "A Security SLA-Driven Moving Target Defense Framework to Secure Cloud Applications," in *Proceedings of the 5th ACM Workshop on Moving Target Defense*, ser. MTD '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 48–56. [Online]. Available: https://doi.org/10.1145/3268966.3268975

[13] R. E. Navas, F. Cuppens, N. Boulahia Cuppens, L. Toutain, and G. Z. Papadopoulos, "MTD, Where Art Thou? A Systematic Review of Moving Target Defense Techniques for IoT," *IEEE Internet of Things Journal*, vol. 8, no. 10, pp. 7818–7832, 2021.

[14] V. Casola, A. De Benedictis, and M. Albanese, *A Multi-Layer Moving Target Defense Approach for Protecting Resource-Constrained Distributed Devices*. Cham: Springer International Publishing, 2014, pp. 299–324. [Online]. Available: https://doi.org/10.1007/978-3-319-04717-1_14

[15] E. Battista, V. Casola, A. Mazzeo, and N. Mazzocca, "SIREN: a feasible moving target defence framework for securing resource-constrained embedded nodes," *International Journal of Critical Computer-Based Systems*, vol. 4, no. 4, pp. 374–392, 2013, pMID: 59053. [Online]. Available: https://www.inderscienceonline.com/doi/abs/10.1504/IJCCBS.2013.059053

[16] I. Zografopoulos, J. Ospina, X. Liu, and C. Konstantinou, "Cyber-physical energy systems security: Threat modeling, risk assessment, resources, metrics, and case studies," *IEEE Access*, vol. 9, pp. 29 775–29 818, 2021.

[17] R. Khan, K. McLaughlin, D. Laverty, and S. Sezer, "STRIDE-based threat modeling for cyber-physical systems," in *2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*, 2017, pp. 1–6.

[18] M. Miettinen, S. Marchal, I. Hafeez, N. Asokan, A. Sadeghi, and S. Tarkoma, "IoT SENTINEL: Automated Device-Type Identification for Security Enforcement in IoT," in *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, 2017, pp. 2177–2184.

[19] M. Ge, J. B. Hong, W. Guttmann, and D. S. Kim, "A framework for automating security analysis of the Internet of Things," *Journal of Network and Computer Applications*, vol. 83, pp. 12–27, 2017.

[20] Z. B. Celik, P. McDaniel, and G. Tan, "Soteria: Automated IoT Safety and Security Analysis," in *2018 USENIX Annual Technical Conference (USENIX ATC 18)*. Boston, MA: USENIX Association, Jul. 2018, pp. 147–158. [Online]. Available: https://www.usenix.org/conference/atc18/presentation/celik

[21] J. Bugeja, B. Vogel, A. Jacobsson, and R. Varshney, "IoTSM: An End-to-end Security Model for IoT Ecosystems," in *2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, 2019, pp. 267–272.

[22] A. Alwarafy, K. A. Al-Thelaya, M. Abdallah, J. Schneider, and M. Hamdi, "A Survey on Security and Privacy Issues in Edge Computing-Assisted Internet of Things," *IEEE Internet of Things Journal*, pp. 1–1, 2020.

[23] V. Casola, A. De Benedictis, C. Mazzocca, and R. Montanari, "Toward automated threat modeling of edge computing systems," in *2021 IEEE International Conference on Cyber Security and Resilience (CSR)*, 2021, pp. 135–140.

[24] K. M. Alam and A. El Saddik, "C2PS: A Digital Twin Architecture Reference Model for the Cloud-Based Cyber-Physical Systems," *IEEE Access*, vol. 5, pp. 2050–2062, 2017.

[25] T. J. Williams, "The Purdue enterprise reference architecture," *Computers in Industry*, vol. 24, no. 2, pp. 141–158, 1994. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0166361594900175

[26] K. Sha, R. Errabelly, W. Wei, T. A. Yang, and Z. Wang, "EdgeSec: Design of an Edge Layer Security Service to Enhance IoT Security," in *2017 IEEE 1st International Conference on Fog and Edge Computing (ICFEC)*, 2017, pp. 81–88.

[27] EU, "Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)," 2016. [Online]. Available: https://eur-lex.europa.eu/eli/reg/2016/679/oj/eng

[28] OWASP, "The Ten Most Critical Web Application Security Risks," 2017.

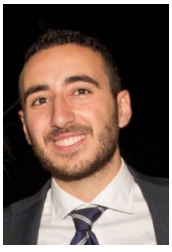
[29] Cloud Security Alliance, "Top Threats to Cloud Computing: Egregious Eleven," 2019. [Online]. Available: https://cloudsecurityalliance.org/artifacts/top-threats-to-cloud-computing-egregious-eleven

[30] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, and B. Sikdar, "A Survey on IoT Security: Application Areas, Security Threats, and Solution Architectures," *IEEE Access*, vol. 7, pp. 82 721–82 743, 2019.

[31] J. Zhang, B. Chen, Y. Zhao, X. Cheng, and F. Hu, "Data Security and Privacy-Preserving in Edge Computing Paradigm: Survey and Open Issues," *IEEE Access*, vol. 6, pp. 18 209–18 237, 2018.
[32] P. Tedeschi and S. Sciancalepore, "Edge and Fog Computing in Critical Infrastructures: Analysis, Security Threats, and Research Challenges," in *2019 IEEE European Symposium on Security and Privacy Workshops (EuroS PW)*, 2019, pp. 1–10.
[33] C. Watson and T. Zaw, "OWASP Automated Threat Handbook," Available at https://github.com/OWASP/www-project-automated-threats-to-web-applications/blob/master/assets/files/EN/automated-threat-handbook-EN-1v20.pdf, 2018.
[34] National Institute of Standards and Technology, "NIST Special Publication 800-53 Revision 4: Security and Privacy Controls for Federal Information Systems and Organizations," 2013. [Online]. Available: http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r4.pdf
[35] C. Sun and L. Zhang, "Design and modeling of intelligent home security monitoring system based on cps," in *2021 IEEE 12th International Conference on Software Engineering and Service Science (ICSESS)*, 2021, pp. 186–189.

Valentina Casola Valentina Casola, Dr. is an Associate Professor at the Department of Electrical Engineering and Informaion Technology of the University of Naples Federico II, Italy. She got a Ph.D. in Computer Engineering from the Second University of Naples in 2004. Her research activities are both theoretical and experimental and focus on security methodologies to design and evaluate distributed systems, including cyber physical infrastructures, cloud systems and web services. These activities are led in cooperation with academic institutions and industrial partners within national and international projects. She has published more than 100 papers in journals, conference proceedings and books.

Alessandra De Benedictis Alessandra De Benedictis received her Ph.D in Computer and Automation Engineering in 2013, both from the University of Naples Federico II, Naples, Italy. She is currently an assistant professor at the Department of Electrical Engineering and Information Technology of the University of Naples Federico II. Her research interests mainly involve the design and evaluation of secure architectures for the protection of distributed systems. She is particularly interested in the definition of methodologies for the development of applications able to offer well-defined security guarantees, both in the cloud environment and in presence of resource constraints. Other relevant research activities include the investigation on moving target defense mechanisms and on embedded security solutions based on reconfigurable hardware.



Carlo Mazzocca received his M.Sc. and B.Sc. degrees in Computer Engineering in 2018 and 2020, respectively, both from the University of Naples Federico II, Italy. He is currently a Ph.D. student in Computer Science and Engineering at the University of Bologna, Bologna, Italy. His research interests mainly include authentication and authorization solutions for cloud continuum systems.

RebeccaMontanari.png

Rebecca Montanari Full professor at the University of Bologna since 2020, she carries out her research in the area of information security and of the design/development of middleware solutions for the provision of services in mobile and pervasive systems, including middleware solutions for new emerging cyber-physical systems and the Internet of Things. Her research is currently focused on the study and design of blockchain technologies to support various supply chains, including agrifood, manufacturing and fashion and on security systems for Industry 4.0.