A Cyber Digital Twin Framework to Support Cyber-Physical Systems Security

Alessandra Somma
University of Naples Federico II
Naples, Italy
Montimage EURL
Paris, France
alessandra.somma@unina.it

Valentina Casola
University of Naples Federico II
Naples, Italy
casolav@unina.it

Ana Rosa Cavalli
Institut Politechnique, Telecom SudParis
Montimage EURL
Paris, France
ana.cavalli@montimage.com

Alessandra De Benedictis
University of Naples Federico II
Naples, Italy
alessandra.debenedictis@unina.it

Wissam Mallouli
Montimage EURL
Paris, France
wissam.mallouli@montimage.com

Valeria Elisa Valdés
Montimage EURL
Paris, France
valeria.valdes@montimage.com

***Abstract*—Cyber-Physical Systems (CPSs) cybersecurity faces several challenges due to the complexity and interconnectedness of involved Information Technology (IT) and Operational Technology (OT) systems, the need to balance safety and security, and the rapid pace of technological change, requiring continuous risk assessment, monitoring, testing and mitigation to ensure their resilience against evolving cyber threats. Unfortunately, due to the criticality of involved systems, some security-related activities such as security testing or patch management may interfere with the correct system operation. Moreover, costs related to security are typically high which may complicate things further. The recent adoption of the emerging Digital Twin technology for cybersecurity purposes allows cybersecurity experts to identify vulnerabilities and develop strategies to prevent and respond to cyber threats in a safe and controlled environment, and in a cost-effective way. The so-called Cyber Digital Twin (CDT) is a cybersecurity-oriented virtual replica of a system, network, or device that can be used, for example, to simulate potential cyber attacks and test security measures.**

A comprehensive CDT framework for CPS systems is presented in this paper, designed to support multiple cybersecurity objectives by suitably extending the core functionalities offered by a generic Digital Twin. To show the effectiveness of our proposal, we discuss the implementation of a proof of concept leveraging the MiniCPS simulator as the core simulation engine and aimed at offering intrusion detection and security testing services.

***Index Terms*—Cyber Digital Twins, Sensors, Modelling, Attack simulations, security controls, Security Services**

I. Introduction

Cyber-Physical Systems (CPSs) integrate cyber and physical processes and components by using advanced computing, networks, and sensor technologies. The increased digitization of modern industrial systems and other Critical Infrastructures (CIs) introduces new attack vectors that may not only put organization's assets at risk, but could also endanger human life. Furthermore, the lack of standardization in CPS security protocols and the shortage of skilled cybersecurity professionals pose additional challenges to securing these systems.

The MITRE Corporation[1] has developed a list of common vulnerabilities and exposures (CVEs) for CPSs, with over 300 entries currently listed. Besides, the Cybersecurity Ventures' report predicts that global cybercrime costs will reach 10.5 trillion euros annually by 2025, with cyberattacks on CIs and Industrial Control Systems (ICSs) expected to be a major driver of this increase[2].

In this scenario, *security testing, monitoring, intrusion detection and risk management* are fundamental activities to carry out. However, due to the criticality of the running systems, testing in the production environment is not always feasible and typically not recommended, as it may negatively impact on the infrastructure. The same criticality can be encountered when applying security patches or configuration changes, which could affect the system unexpectedly. On the other hand, the setup and maintenance of test environments and the implementation and deployment of security countermeasures are expensive.

The concept of *Digital Twin* (DT) has been around for more than 10 years now, and it is currently witnessing a great popularity thanks to its adoption in several industrial and research fields [1]. A DT is a dynamic and self-evolving digital replica of an object/system/process characterized by a seamless connection between the physical and the virtual worlds established through the exchange of real-time data generated by the physical replica when suitably instrumented. Recently, the DT concept is being increasingly associated with cybersecurity [2]–[6], as the digital replica of a system can be

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

[2] https://www.esentire.com/resources/library/2022-official-cybercrime-report

used to simulate and test potential cyber attacks and to evaluate the effectiveness of cybersecurity measures.

By creating the so-called **Cyber Digital Twin** (CDT) [7] of a CPS, it is possible to overcome the limitations discussed above and enable the execution of security activities such as security testing, training, attack and anomaly detection, security measures definition and optimization in a safe and controlled environment and in a cost-effective way. This approach allows organizations to stay ahead of evolving cyber threats, improve their cybersecurity posture and ensure the safety and security of their CPS. Despite the large interest in CDTs and the availability of several solutions in different domains, we have observed that often these proposals lack a formalization from the architectural point of view. The distinctive characteristics of a DT sometimes get lost, even because existing CDT solutions typically only model specific security-related aspects (e.g., a set of attack techniques) that are often reduced to mere security techniques. Moreover, other aspects of the system not related to security that may be useful/needed to acquire a comprehensive system view are completely left out from virtualization.

In light of the above considerations, we believe that the definition of a structural or functional breakdown of a CDT, possibly identifying and connecting the core modules/functionalities needed to offer security-related services, would sensibly improve the uptake of this technology and its understanding. Hence, in this paper, we present a CDT Framework for CPSs which is aimed to support multiple cybersecurity properties by suitably extending the core functionalities offered by a generic Digital Twin. To show the effectiveness of our proposal, we discuss the implementation of a proof of concept (PoC) leveraging the MiniCPS[3] [8] simulator as the core simulation engine and aimed at offering intrusion detection, security testing and security controls optimization services.

The paper is organized as follows: in Section II, we provide an overview of the recent literature on CDTs to identify supported security objectives and to highlight the current lack of architectural proposals. In Section III, we present our CDT Framework, illustrating devised layers and functionalities. In Section IV, we describe and discuss our proof of concept CDT by outlining the mapping of its modules and components onto our framework functionalities. Finally, in Section V we draw our conclusions.

II. CDTs APPLICATIONS AND ARCHITECTURAL SOLUTIONS: RELATED WORK

A. CDT security objectives

As pointed out by recent studies [2]–[6], the DT technology is being increasingly investigated as a means to improve cybersecurity of cyber-physical systems, such as ICSs and smart grids. Several approaches have been proposed in the literature, which enable to identify a set of different (while converging) cybersecurity objectives that a DT can help pursue. Identified security objectives mainly belong to the following categories:

- **Security testing**: DTs may be used during system operation to perform non-invasive penetration testing activities without causing disruption or damage to services and equipment [9]–[11].
- **Cybersecurity training**: DTs may be used to build cyber ranges and help personnel train on cybersecurity issues and solutions [12], [13].
- **Attack/intrusion/anomaly detection**: real-time inputs may be processed by the virtual replica to discover ongoing or upcoming cyberattacks/intrusion attempts through simulation. Alternatively, the behavior or the status of the virtual replica may be compared with that observed from the actual system to discover anomalies due to cyber attacks [14], [15]; this includes also the detection of SW/HW misconfigurations [16] which may be indicative of malicious manipulation, as well as the detection of privacy issues by combining and correlating privacy data and assessing the compliance to regulations (i.e., GDPR) [17].
- **Security controls selection**: a DT may be used to evaluate in advance the impact on a system of the enforcement of specific security controls (and of related implementation solutions) before their actual deployment. This activity may be carried out based on a risk analysis and evaluation step, which could help developers prioritize security controls implementation to optimize cost and maximize benefits [7], [18]. The CDT objective of evaluating the impact of a security control before its actual enforcement in a system is sometimes referred to a **security patch management**, especially when considering Operational Technolgy (OT) systems [6], where the application of a security patch would likely impact the whole OT infrastructure and must be carefully evaluated beforehand. The possibility to evaluate security controls' implementation in advance is particularly useful in case of particularly costly resiliency strategies [19] that involve for example the deployment of different replicas of a component or system or the implementation of diversity techniques.
- **Cyber-Threat Intelligence (CTI) generation**: CDT may be used to generate and share CTI data based on the simulation of planned incident scenarios [20].
- **Diagnostics**: based on historical data collected and analysed by the CDT, it is possible to perform diagnostic activities aimed at investigating the causes of incidents [15].

B. CDT architectural solutions

Of existing approaches, only a few present an architectural solution and/or an implementation. Among these, the authors of [14] discuss the implementation of an IoT-based DT of

[3] https://github.com/scy-phy/minicps

a system of cyber-physical networked microgrids, able to detect coordinated false data injection and denial of service attacks. While they provide a detailed mathematical model of both the physical microgrids and the control system and also discuss a cloud-based implementation, the overall DT architecture is only sketched at high-level and core functionalities/components are not explicitly identified.

A more detailed architecture is provided by Hadar *et al.* in [7], where a proprietary cyber digital twin framework is presented, aimed at identifying and evaluating existing cybersecurity risks and at supporting the identification and prioritization of security controls. The approach leverages the automatic generation of Analytical Attack Graphs (AAG), which model attack behavior based on the knowledge on the system's assets and on the adoption of logical rules derived by MITRE ATT&CK[4] tactics. Based on generated attack graphs, the approach proposed in [7] enables to simulate the behavior of the system when a set of security controls is in place, and to evaluate the impact of such controls on the organization's overall cyber risk exposure. This can be used to prioritize controls based on expected costs and risk reduction potential. Although the proposed architecture is proprietary, the work is useful to identify a set of functionalities that may be offered as building blocks for advanced security services, hopefully exploiting open knowledge bases and resources. These functionalities include assets and vulnerabilities discovery, attack and defense modeling, threat intelligence, risk evaluation, countermeasure identification and prioritization.

Atalay *et al.* [9] propose a DT-based approach aimed at enabling non-intrusive security testing of smart grids. The authors describe a three-layer architecture featuring a physical, a virtual, and a decision-making layer, the latter being composed of algorithms mapping the physical tiers to the virtual tiers, optimizing monitored data to improve the health of the system, and raising alarms during anomalous situations. The authors identify some core elements in the system represented by a CTI database (for grid-specific attacks and other common attacks on the application and network layers), an attack simulation tool set including implementations of the attacks in the threat intelligence database, and a data analysis and reporting module that makes vulnerability inferences and risk assessment based on the results of attack simulations performed on the DT.

Eckhart and Ekelhart in [16] propose an approach to automatically generate the DT of a CPS based on a formal specification. The generated DT can be enriched with the integration of specific modules that enable to build security features such as monitoring, intrusion detection and anomaly detection. Security and safety analysis is carried out based on suitable rules that are codified in the specification and which represent conditions that must hold (e.g., two velocity parameters must be equal). Although the authors explicitly refer to an architectural solution and also provide a proof of concept for a simple ICS modeled with AutomationML, the proposed architecture is quite high-level, and does not sufficiently break-up needed capabilities to achieve different security objectives.

Empl *et al.* in [15] discuss the integration of security data analytics capabilities within digital twins, and explicitly distinguish among descriptive, diagnostic, detective, predictive, and prescriptive operations. A high-level architectural model is considered that simply identifies physical and virtual environments hosting respectively the physical systems and its digital representation, and a security analytics layer fed by the latter. The virtual environment includes the digital twin and its data, including descriptive and dynamic asset data, dynamic environment data, historical asset data, and semantics.

III. Proposal of a Cyber Digital Twin platform

From our review of the recent literature on CDTs discussed in Section II, it is clear that the adoption of the DT technology for cybersecurity purposes is very promising, and will likely attract more and more attention in next years from the scientific community working on the different aspects of the cybersecurity risk management cycle. However, it must be noted that, despite the availability of a few implementations in different domains (e.g., IoT, ICSs, microgrids), proposed solutions typically do not explicitly show how the techniques used to accomplish the considered cybersecurity objectives (e.g., penetration testing or attack detection) are integrated into the "underlying" DT. In other words, the DT concept is often used as a "new" term to re-baptize something not new at all. This is even more true if considering that, in most of existing proposals, the CDT only models security-related behavior and no other aspect of the system is taken into account. In regard to this consideration we must note that, at the best of our knowledge, no attempt to define a comprehensive CDT architecture has been made. We believe that the definition of a structural or functional breakdown of a CDT, possibly identifying and connecting the core modules/functionalities needed to offer security-related services, would sensibly improve the uptake of this technology.

In light of the above considerations, we propose a CDT platform that clearly identifies and integrates core DT functionalities with security-related functionalities, and organizes them in a set of logical layers. The platform, schematically depicted in Fig. 1, extends the baseline DT architecture proposed in [21] with specific layers and functionalities dedicated to cybersecurity.

In particular we devise five layers, including (i) the **Physical Twin layer**, which hosts the instrumented CPS, (ii) the **Digital Twin layer**, which hosts the virtual replica of the system, and (iii) the **Service layer**, which offers the different security-related services, plus two data layers, namely (iv) the **PT-DT Data layer**, which manages the data generated from the

[4] https://attack.mitre.org/

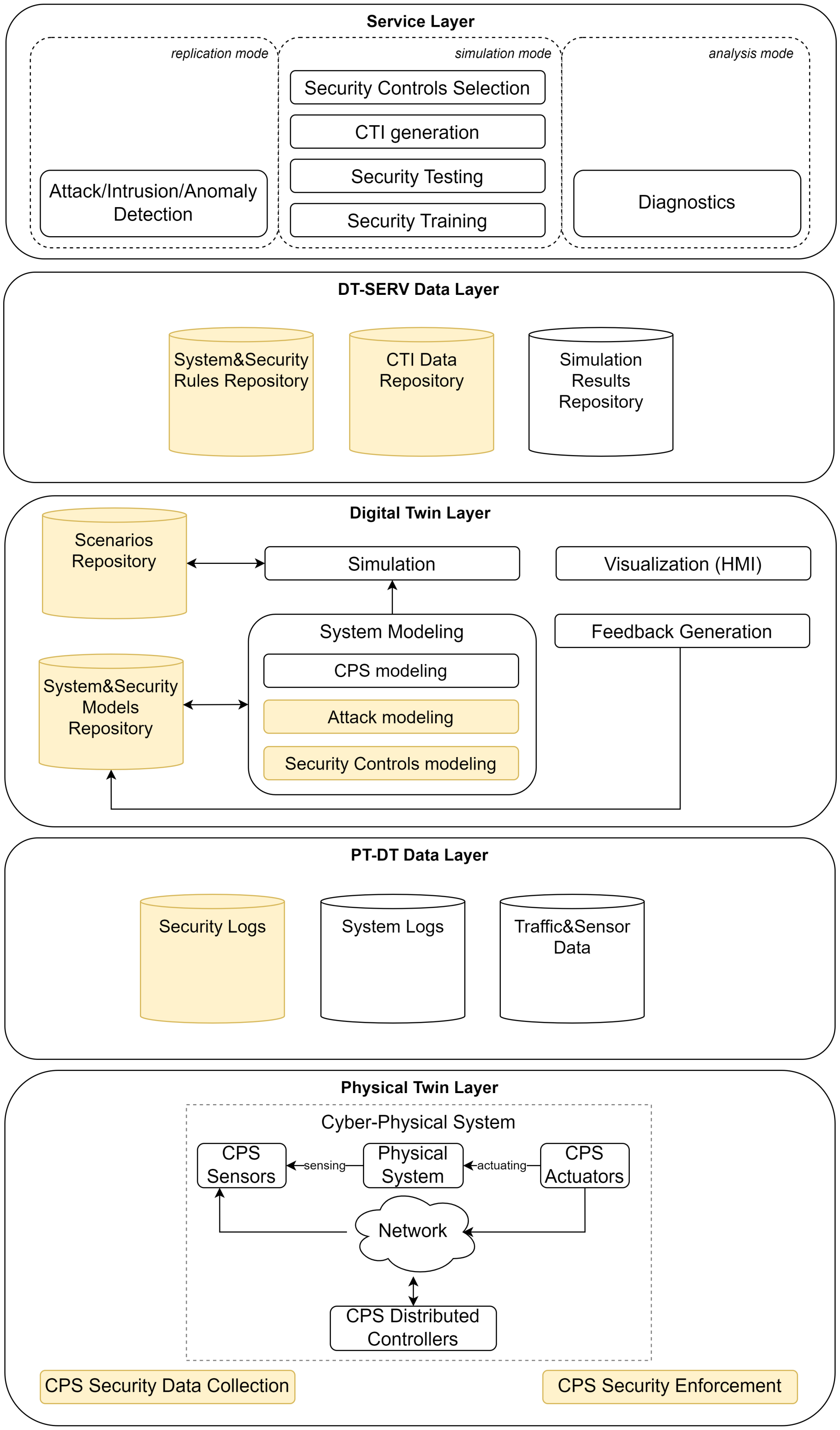

Fig. 1. CDT Platform

Physical Twin layer, and (v) the **DT-SERV Data layer**, which manages the data generated by the Digital Twin layer.

The Physical Twin layer includes the original CPS along with all sensors and actuators needed respectively to collect data from the CPS in order to build its DT, and to control the CPS based on possible DT-based feedback. In this layer, we explicitly include a *CPS Security Data Collection* functionality, represented by the deployment of security probes and agents able to collect security-related data (e.g., Indicator of Compromise) and other parameters linked to security metrics [22]. Also, we explicitly add a *CPS Security Enforcement* functionality, which consists in the actual deployment and activation of security controls as result of analyses conducted thanks to the services offered by the Security layer. Security enforcement within the CPS may require an additional development and testing stage or, when possible, it may leverage automated mechanisms such as those presented in [23].

All the data generated by the Physical Twin layer (both security-related and not security-related) are managed within the PT-DT Data layer and are exploited by the Digital Twin layer to actually build a virtual representation of the system. PT-DT data include security logs as well as (aggregated and filtered) sensor data and other system logs, regarding for example the operations executed by the CPS controller subsystems. This data can be used for diagnostic purposes to identify the source or cause of a security incident.

The Digital Twin layer includes four main functionalities, namely *System Modeling, Simulation, Feedback Generation* and *Visualization*. System Modeling includes activities performed to statically or dynamically build a comprehensive model of the CPS, able to capture functional and security-related aspects. This functionalities relies upon a System&Security Models Repository that manages all available models. Several modeling approaches exist both for representing CPSs' assets and functionalities and for representing security properties. AutomationML[5] for example is a popular XML-based language that can be used to describe complete engineering chain of production systems by offering a standardized data exchange format. It has been used to generate automatically a CDT of a CPS from its specification [16]. With regard to security modeling, attack graphs, attack trees and Petri Nets are the most popular tools. Attack graphs can be used to model adversarial behavior taking into account actual assets and related vulnerabilities and have been widely used to simulate attacks and support security testing [2].

In fact, the modeling functionality is directly exploited by *Simulation*, which enables to "execute" models based either on pre-determined scenarios (available in a Scenarios repository) or based on real-time data suitably fed by the underlying level. Finally, the *Feedback Generation* functionality enables to trigger updates both in the Physical Twin layer (for example, by launching the enforcement of specific countermeasures) and in the system models accordingly. To conclude, *Visualization* represents the traditional DT function to report system present and future status to human users to help them taking decisions and possibly providing feedback.

Above the Digital Twin layer, we devised another data management layer that collects specific knowledge needed to build upper services, i.e., the DT-SERV layer. This includes Systems&Security Rules that can be used for example for attack/intrusion/anomaly detection, and CTI data that can be used to perform security testing. Moreover, this layer manages simulation results, which are used in most security services.

In the Service layer we included the main services identified from our literature review. In particular, we classified these services based on the DT operational mode exploited. In fact, as outlined in [15], DTs can be executed in different operational modes, i.e., simulation, analysis, and replication based on what kind of data they use. In analysis mode, historical data are analysed with statistical means to extract knowledge and perform, for example, diagnostics and root-cause analysis. In simulation mode, specification data taken from models are used to simulate or emulate the behavior of the system: this is useful to perform testing, training, generation of CTI data and evaluation of security controls in place. Finally, in replication mode, the DT is used online and models are fed with real-time data: this is useful for detection purposes.

IV. Proof of concept: using the CDT framework to build an intrusion detection and security testing service

In this Section, we discuss a PoC consisting in the construction of a CDT devoted to offering intrusion detection, security testing and security controls selection services based on the proposed CDT Framework functionalities. The PoC has been implemented by leveraging a popular simulation engine for CPS named MiniCPS [6] [8]. Before presenting the architecure of the PoC, we will provide the required background on MiniCPS functionalities and architecture as it will be explicitly mapped onto our framework.

A. PoC background: MiniCPS

MiniCPS is a framework for CPSs real-time simulation that aims to create an extensible, reproducible environment for network communications, control systems and physical-layer interactions in CPS, allowing researchers to emulate the Ethernet-based network of Industrial Control Systems (ICS). It is built on top of *Mininet*[7], a network emulator, that, through lightweight virtualization, allows to emulate a collection of end-hosts, i.e., switches, routers, middle boxes, and links with a high level of fidelity [24]. Each virtual host is a collection of processes isolated into a container and links are emulated using virtual Ethernet (`veth`), thus the network communication in MiniCPS uses the default Linux networking stack.

MiniCPS offers a microservices-based architecture in which each component of the system is realized using Docker containers. In the top layer there is the *network* through which

[5] https://www.automationml.org/

[6] https://github.com/scy-phy/minicps

[7] https://github.com/mininet/mininet

messages are exchanged. Thanks to Mininet, all standard protocols (e.g., ICMP, HTTP, NTP, etc.) can be used; in particular, MiniCPS uses the CPPPO Python library to provide EtherNet/IP (ENIP) services. *CPS components* are connected to the network and their implementation is done through simple scripts. The simulated components can access specific *Physical Layer* properties through the *Physical Layer API*: this allows the interaction between CPS simulated components and their real world counterpart [8].

MiniCPS has been used to implement CDTs. First, Dietz *et al.* [25] integrated the Digital Twin of an industrial filling plant, implemented as a standalone simulation using Mininet-based MiniCPS, with a security operation center[8]. Then, Varghese *et al.* [26] extended this tool with a simulator of different types of process-aware attacks, i.e., command injection, network Denial of Service (DoS), calculated measurement modification and naive measurement modification, and an Intrusion Detection System (IDS) module based on Machine Learning (ML), using a stacked ensemble classifier model[9]. As the case study that we will present in Section IV will relay on Varghese *et al.* [26] digital-twin based security framework for ICSs, here we briefly describe the simulated physical system and its network topology.

The industrial filling plant consists of a *tank* and an actuator, e.g., a motor-driven valve (*Actuator1-MV*) that controls the outflow of the fluid from the tank to the bottle through a pipe. The tank and the bottle are equipped with a liquid level (LL) sensor, i.e., *Sensor1-LL* and *Sensor3-LL*, while on the pipe there is installed a flow level (FL) sensor, i.e., *Sensor2-FL*. Three Programmable Logical Controllers (PLCs) monitor the sensors and the actuator: *PLC1* controls the sensor measuring the tank liquid level (Sensor1-LL) and executes the control strategy (open/close) of the motor-driven valve (Actuator1-MV). As the controlling activity depends on the other sensors values, PLC1 receives the other sensors values, Sensor2-FL and Sensor3-LL, managed by *PLC2* and *PLC3* respectively. Sensor measurements are stored as ENIP tags in corresponding PLCs, e.g., "Sensor2-FL" tag is stored in PLC2 and can be requested and received by PLC1. Fig. 2 depicts the network topology of the industrial filling plant: apart from the three PLCs, there are a *Human Machine Interface* (HMI) that allows direct control of the actuator and the *attacker*; each component communicates through the router. Please, note that the attacker is decided to be placed inside the network of the simulated industrial filling plant, thus inside the Digital Twin because of security use cases illustrations [25], [26]. However, in real world scenarios, the DT runs in isolated and secure environments, this means that unauthorized accesses are not allowed.

The extension of the framework proposed by Varghese *et al.* [26] models and executes different process-aware attack scenarios as insider threats in the digital twin container: attacks interfere with physical processes in real life, thus process measurements collected from the digital twin mirror the consequences of attacks occurring in the physical system. More in detail, in Varghese *et al.* [26] work, authors implemented 23 attack scenarios for four different attack types:

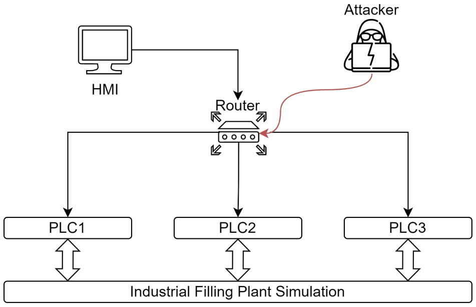

Fig. 2. Network topology of the simulated industrial filling plant: three PLCs managing sensors and actuator, HMI and attacker [25].

- *Command injection* remotely inserts malicious commands in order to control the motor valve exploiting the PLC1 interface.
- *Network DoS* prevents the PLC1 from receiving any or specific (e.g., from PLC2 and so from Sensor2-FL) measurements.
- *Calculated measurement modification* is a false data injection/modification attack that exploits the lack of encryption in the ENIP protocol and alters PLC1 data to calculated values.
- *Naive measurement modification* is similar to the previous one, except that the sensor measurements are altered to constant values.

Finally, they added the IDS module as a Docker container, implementing and evaluating supervised ML algorithms using the Scikit-learn[10] Python library.

As already remarked, these works as the others are missing an architectural model that can help us implement CDT platforms in order to offer different kinds of security services (e.g., security testing or security control identification).

B. PoC Description

As anticipated, we leveraged the MiniCPS toolkit to build our PoC. In particular, we exploited the modules offered by the MiniCPS extension including the IDS and the attacks simulator to provide some of the functionalities proposed in our CDT framework. To test a relevant scenario, we implemented and integrated within MiniCPS a specific security control that represents one of the most common CPS resilience strategies [27], i.e., ***redundancy***, in order to offer the *security control selection* service, aimed at evaluating the effects of the enforcement of a given control within the system. Moreover,

[8] https://github.com/FrauThes/DigitalTwin-SIEM-integration
[9] https://github.com/sebavarghese/DT-based-IDS-framework
[10] https://scikit-learn.org/stable/

we implemented specific attacks to test the effectiveness of this countermeasure to demonstrate the construction of a *security testing service*. Finally, we arranged the built-in IDS to fit into an *intrusion detection service* offered on top of the CDT.

Fig. 3 shows the platform architecture of our case study with respect to the general one depicted in Fig. 1. The Physical Twin Layer is characterized by the components of the already presented industrial filling plant with the addition of the `Monitor` that enables the CPS security data collection in the real world. The PT-DT Data Layer contains `PLCs logs data` obtained (that contain also sensors data) and `Monitor log`, namely data received from the `Monitor` component. In the Digital Twin Layer there are: i) physical twin components modeled with `MiniCPS`; ii) physical twin security `Monitor`; iii) `attacks models`; iv) `security controls models` (i.e., redundancy). Each of these models is a Python file. Moreover, `XTerm`[11] enables the simulation of every modeled element (apart from attacks) but also the human machine interfacing. In fact, it is possible to view real time data on `XTerm` shells and interact through the command line. Instead, modeled attacks are executed through `Ettercap`[12] and `Hping3`[13] simulators. The `feedback` component is needed to interact with the physical system and send commands according to simulation results: as we are executing the Cyber Digital Twin in simulation mode, thus without connecting it to a real replica, no feedback is implemented. The DT-SERV layer contains `PLCs logs data` obtained from simulation and `security datasets`, i.e., one for training the intrusion detection system and the other one is real-time generated during security simulations. Finally, the *Service Layer* includes the ML-based `intrusion detection` system and the `incident detection` (with DSIEM tool[14]) already existing in [25], [26]. We add the `security testing` module for testing the redundancy strategy, but other ones can be added. This would enable the `security control identification` service: the security technique that, executed in DT layer, performs in the best manner from a security perspective, can be implemented in the PT layer through *Feedback* and *CPS security enforcement* modules, shown in the general architecture.

In order to evaluate the redundancy technique in a CDT platform, first of all we have to modify the network topology, as depicted in Fig. 4: in particular, we decide to ***replicate the PLC1 component*** two times to cope with attacks pursued against this controller that manages not only data coming from all the sensors but also the working activity of the actuator. Duplicating PLC1 means that $PLC1\_R1$ and $PLC1\_R2$ present the same behaviour and their implementations are equals, apart from their MAC and IP addresses.

The ***load balancer*** manages the switching between the two replicas according to a certain prefixed time window, *e.g.*, every 2 minutes switches from $PLC1\_R_1$ to $PLC1\_R_2$ or from $PLC1\_R_2$ to $PLC1\_R_1$. Note that $PLC1\_R_1$ and $PLC1\_R_2$ are two **living replicas**: this means that both of them receive inputs from the industrial filling plant in order to process them and calculate the actuator command according to the actuating law of the physical system, but just the active replica in that time window can send the command to the actuator. In other words, every replica receives inputs and identifies the value to be sent to the actuator, but in order to correctly implement the redundancy technique the PLC_R_i sends the calculated actuator command to the load balancer and then the load balancer forwards the command received by the active PLC_R_i to the motor-driven valve of the industrial filling plants. This explains why, in Fig. 4, there are the bidirectional arrow that connects the PLC1s to the physical system and the arrow that connects the load balancer to the simulated physical plant. Moreover, the two replicas are **independent**, thus there is no communication between them. This means that an attacker may attack the communication between one replica and one or more PLCs, but not both of them. Clearly, if there is more than one attacker, it is possible to attack both replicas and the whole system will not work anymore.

C. Experimental setup and evaluation result

The framework runs on a Linux (Ubuntu 22.04) virtual machine (i.e., VirtualBox) hosted on a Microsoft Windows 11 machine. We test the redundancy in four different cases, i.e., when the load balancer switches every $\{2, 5, 10, 20\}$ milliseconds. Every simulation has been run for 30 minutes in which every six minutes we attacked the $PLC1\_R_1$ for six minutes. Each table reports the results of experiments for the different kinds of simulated attacks (i.e., Network DoS, Command Injection): columns represent the time window values, while rows indicate the *number of attacks* received by $PLC1\_R_1$, the *number of received attacks* by the whole system (i.e., the number received by the load balancer when the $PLC1\_R_1$ is active) and the *number of detected attacks* as we are doing detection through ML-based IDS of Varghese *et al.* tool [26], thus we expect to have the same accuracy, namely using the stacked model 100% for Network DoS and 91% for command injection.

We can easily understand when the $PLC1\_R_1$ is under attack thanks to the "*PLC1_R1.log*" file. For instance, if we are pursuing a MITM/DoS (Man-in-the-middle/Denial of Service) attack, e.g. with the command:

```
ettercap -T -i attacker-eth0 -M ARP
     /10.0.0.1// /10.0.0.2//
```

that generates attack between $PLC1\_R_1$ and $PLC2$, the log file presents the following warning:

[11] http://xtermjs.org/

[12] https://www.ettercap-project.org/

[13] https://www.kali.org/tools/hping3/

[14] https://www.dsiem.org/

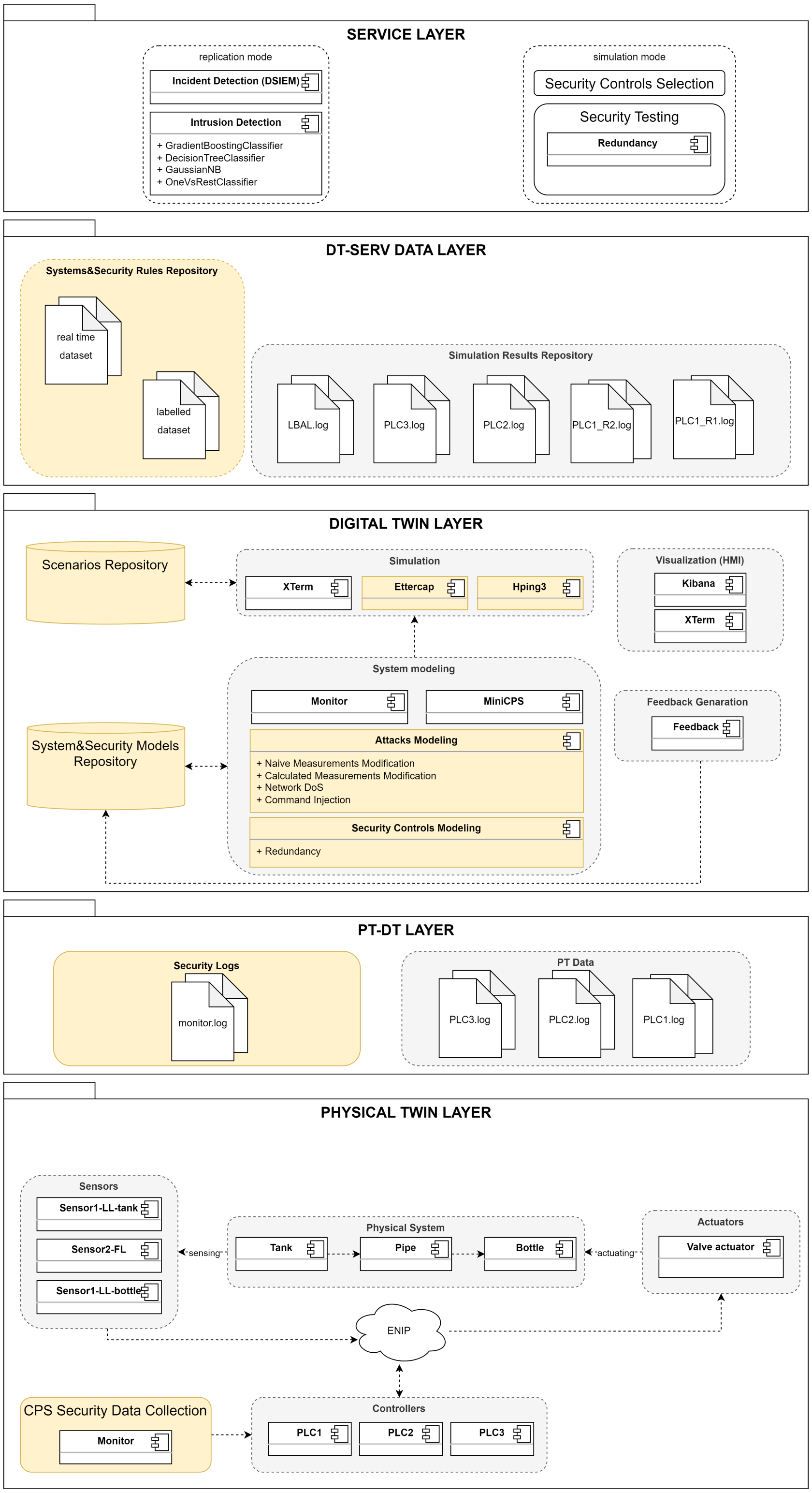

Fig. 3. Case study platform architecture with respect to the general one presented in Section III.

```
WARNING 03/15/2023 11:19:19 ``10.0.0.1
main_loop Flow level (SENSOR 2) is not
received. Program is unable to proceed
properly.''
```

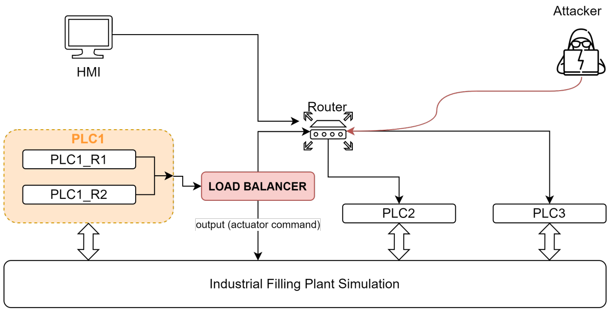
Attacker
HMI
Router
PLC1
PLC1_R1
PLC1_R2
LOAD BALANCER
output (actuator command)
PLC2
PLC3
Industrial Filling Plant Simulation

Fig. 4. Modified network topology of the simulated industrial filling plant.

TABLE I
Results of Experiments (MITM/DoS attacks).

	2	5	10	20
attacks to $PLC1\_R_1$	152	161	110	78
received attacks by $LBAL$	80	110	90	70
detected attacks by $ML-IDS$	79	110	90	70

TABLE II
Results of Experiments (Command injection attacks).

	2	5	10	20
attacks to $PLC1\_R_1$	148	159	104	100
received attacks by $LBAL$	79	108	93	95
detected attacks by $ML-IDS$	71	99	85	87

and the flow level will be set up at 999 (out of range value). From "*LBAL.log*" (generated by the load balancer), we can understand which of the two replicas is active:

```
INFO 03/15/2023 11:19:19 ``10.0.0.7
main_loop Executing replica #N: 1.''
```

Finally, through the simulation and monitoring activities, we can classify the real time dataset and understand when the system is detected to be under attack:

```
15/03/2023 11:19:19 3,37269E+15 999
      6,35185E+14 Network DoS
```

Table I shows what happens in case of *Denial of Service* attacks, instead in Table II there are the results in case of *Command Injection* attacks. As expected, frequently switching between the two replicas allows to decrease the probability that the whole system would be under attack when the $PLC1\_R_1$ is attacked. Moreover, we notice how the obtained results of attack detection are compliant with Varghese *et al.* [26] ones.

D. Discussion

We demonstrate the application of the general CDT platform framework using a toolkit for security research, i.e., MiniCPS enriched with incident and intrusion detection. Moreover, we model and execute a security countermeasure in a CDT environment: we find out that reducing the switching time window between PLC1 replicas ensures a better reaction and protection against DoS and command injection attacks. Clearly, we consider just the two-time replication, but it would be possible also to test redundancy increasing and/or decreasing the number of replicas.

However, even if MiniCPS represents a useful tool for research, it seems to be quite difficult to apply it for modeling and simulation of different kind of countermeasures in order to choose the best one: i) adding and/or removing components from the simulated physical twin layer requires a lot of effort; ii) as pointed out by authors [8], MiniCPS is not a performance simulator (in fact, simulation is expensive from a computational resources point of view and it is also really slow) and not even a tool for optimization. Moreover, MiniCPS allows the modeling of only ENIP communication based physical system. The extension of MiniCPS proposed by [25], [26] enables the attacks simulation assuming that the attacker knows the ICS process and architecture and this is not always true.

Finally, our case study executes the CDT in a simulation mode, thus there is no real twin to which the digital replica is connected. Clearly, in order to better prove the feasibility and applicability of our proposal and to evaluate performance of setting up the whole platform for testing different countermeasures, we plan to connect the DT to its physical counterpart and implement other kind of CPS security and resilience enforcement solutions, e.g., moving target defense.

V. Conclusions and Future work

Cyber Digital Twin technology, i.e., digital twin technology for cybersecurity, has the potential to revolutionize the way we design, operate, and maintain complex critical systems. Providing a virtual model of a physical system allows to optimize security, reduce costs and minimize downtime. As more and more industries adopt this technology, we can expect to see significant advancements in security efficiency, data privacy, and system sustainability.

In this paper, we propose a 5-layered CDT framework tackling the main potential security services that we can find in a real system (e.g., security testing, intrusion detection, countermeasure selection and optimization). We instantiate this architecture to build a proof-of-concept CDT devoted to offering security testing and intrusion detection services and implement them by leveraging a popular simulation engine for CPS, called MiniCPS. This experimentation shows the effectiveness of the proposed architecture that can be generalized for other security usage.

As future work, we plan to apply this architecture to a real complex industrial system in the energy domain and check the complexity of such architecture implementation dealing with heterogeneous physical devices requiring different modelling techniques. Besides, as with any emerging technology, there are also concerns around data security, privacy, and ethical implications that must be carefully addressed to ensure its safe and responsible use in a real industrial context.

VI. Acknowledgment

Funded by the European Union

DYNABIC

This work is partially supported by the European Union's Horizon Europe research and innovation program under grant agreement No 101070455 (DYNABIC). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

References

[1] M. Attaran and B. G. Celik, "Digital twin: Benefits, use cases, challenges, and opportunities," *Decision Analytics Journal*, vol. 6, p. 100165, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S277266222300005X

[2] T. Zheng, M. Liu, D. Puthal, P. Yi, Y. Wu, and X. He, "Smart grid: Cyber attacks, critical defense approaches, and digital twin," 2022. [Online]. Available: https://arxiv.org/abs/2205.11783

[3] G. Lampropoulos and K. Siakas, "Enhancing and securing cyber-physical systems and industry 4.0 through digital twins: A critical review," *Journal of Software: Evolution and Process*, vol. e2494, 2022. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/smr.2494

[4] M. Eckhart and A. Ekelhart, *Digital Twins for Cyber-Physical Systems Security: State of the Art and Outlook*. Cham: Springer International Publishing, 2019, pp. 383–412. [Online]. Available: https://doi.org/10.1007/978-3-030-25312-7_14

[5] A. De Benedictis, C. Esposito, and A. Somma, "Toward the adoption of secure cyber digital twins to enhance cyber-physical systems security," in *Quality of Information and Communications Technology*, A. Vallecillo, J. Visser, and R. Pérez-Castillo, Eds. Cham: Springer International Publishing, 2022, pp. 307–321.

[6] D. Holmes, M. Papathanasaki, L. Maglaras, M. A. Ferrag, S. Nepal, and H. Janicke, "Digital twins and cyber security – solution or challenge?" in *2021 6th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM)*, 2021, pp. 1–8.

[7] E. Hadar, D. Kravchenko, and A. Basovskiy, "Cyber digital twin simulator for automatic gathering and prioritization of security controls' requirements," in *2020 IEEE 28th International Requirements Engineering Conference (RE)*, 2020, pp. 250–259.

[8] D. Antonioli and N. O. Tippenhauer, "Minicps: A toolkit for security research on cps networks," in *Proceedings of the First ACM Workshop on Cyber-Physical Systems-Security and/or PrivaCy*, ser. CPS-SPC '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 91–100. [Online]. Available: https://doi.org/10.1145/2808705.2808715

[9] M. Atalay and P. Angin, "A digital twins approach to smart grid security testing and standardization," in *2020 IEEE International Workshop on Metrology for Industry 4.0 & IoT*, 2020, pp. 435–440.

[10] R. Bitton, T. Gluck, O. Stan, M. Inokuchi, Y. Ohta, Y. Yamada, T. Yagyu, Y. Elovici, and A. Shabtai, "Deriving a cost-effective digital twin of an ics to facilitate security evaluation," in *Computer Security*, J. Lopez, J. Zhou, and M. Soriano, Eds. Cham: Springer International Publishing, 2018, pp. 533–554.

[11] E. W. Van der Wal and M. El-Hajj, "Securing networks of iot devices with digital twins and automated adversary emulation," in *2022 26th International Computer Science and Engineering Conference (ICSEC)*, 2022, pp. 241–246.

[12] A. Bécue, Y. Fourastier, I. Praça, A. Savarit, C. Baron, B. Gradussofs, E. Pouille, and C. Thomas, "Cyberfactory#1 — securing the industry 4.0 with cyber-ranges and digital twins," in *2018 14th IEEE International Workshop on Factory Communication Systems (WFCS)*, 2018, pp. 1–4.

[13] A. Bécue, M. Praddaude, E. Maia, N. Hogrel, I. Praça, and R. Yaich, "Digital twins for enhanced resilience: Aerospace manufacturing scenario," in *Advanced Information Systems Engineering Workshops*, J. Horkoff, E. Serral, and J. Zdravkovic, Eds. Cham: Springer International Publishing, 2022, pp. 107–118.

[14] A. Saad, S. Faddel, T. Youssef, and O. A. Mohammed, "On the implementation of iot-based digital twin for networked microgrids resiliency against cyber attacks," *IEEE Transactions on Smart Grid*, vol. 11, no. 6, pp. 5138–5150, 2020.

[15] P. Empl and G. Pernul, "Digital-twin-based security analytics for the internet of things," *Information*, vol. 14, no. 2, 2023. [Online]. Available: https://www.mdpi.com/2078-2489/14/2/95

[16] M. Eckhart and A. Ekelhart, "Towards security-aware virtual environments for digital twins," in *Proceedings of the 4th ACM Workshop on Cyber-Physical System Security*, ser. CPSS '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 61–72. [Online]. Available: https://doi.org/10.1145/3198458.3198464

[17] V. Damjanovic-Behrendt, "A digital twin-based privacy enhancement mechanism for the automotive industry," in *2018 International Conference on Intelligent Systems (IS)*, 2018, pp. 272–279.

[18] M. Masi, G. P. Sellitto, H. Aranha, and T. Pavleska, "Securing critical infrastructures with a cybersecurity digital twin," *Software and Systems Modeling*, vol. 2023, 2023.

[19] B. Cassottana, M. M. Roomi, D. Mashima, and G. Sansavini, "Resilience analysis of cyber-physical systems: A review of models and methods," *Risk Analysis*, vol. n/a, no. n/a. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/risa.14089

[20] M. Dietz, D. Schlette, and G. Pernul, "Harnessing digital twin security simulations for systematic cyber threat intelligence," in *2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC)*, 2022, pp. 789–797.

[21] A. De Benedictis, F. Flammini, N. Mazzocca, A. Somma, and F. Vitale, "Digital twins for anomaly detection in the industrial internet of things: Conceptual architecture and proof-of-concept," *IEEE Transactions on Industrial Informatics*, pp. 1–11, 2023.

[22] V. Casola, A. De Benedictis, M. Rak, and U. Villano, "A security metric catalogue for cloud applications," in *Complex, Intelligent, and Software Intensive Systems*, L. Barolli and O. Terzo, Eds. Cham: Springer International Publishing, 2018, pp. 854–863.

[23] V. Casola, A. De Benedictis, M. Eraşcu, J. Modic, and M. Rak, "Automatically enforcing security slas in the cloud," *IEEE Transactions on Services Computing*, vol. 10, no. 5, pp. 741–755, 2017.

[24] B. Lantz, B. Heller, and N. McKeown, "A network in a laptop: Rapid prototyping for software-defined networks," in *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, ser. Hotnets-IX. New York, NY, USA: Association for Computing Machinery, 2010. [Online]. Available: https://doi.org/10.1145/1868447.1868466

[25] M. Dietz, M. Vielberth, and G. Pernul, "Integrating digital twin security simulations in the security operations center," in *Proceedings of the 15th International Conference on Availability, Reliability and Security*, ser. ARES '20. New York, NY, USA: Association for Computing Machinery, 2020. [Online]. Available: https://doi.org/10.1145/3407023.3407039

[26] S. A. Varghese, A. Dehlaghi Ghadim, A. Balador, Z. Alimadadi, and P. Papadimitratos, "Digital twin-based intrusion detection for industrial control systems," in *2022 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, 2022, pp. 611–617.

[27] B. Cassottana, M. M. Roomi, D. Mashima, and G. Sansavini, "Resilience analysis of cyber-physical systems: A review of models and methods," *Risk Analysis*, 2023-01-16.