DOI: 10.1109/CSR57506.2023.10224997

Threat intelligence using Digital Twin honeypots in Cybersecurity

Maria Nintsiou
Department of Electrical and Computer Engineering
University of Western Macedonia
Kozani, Greece
mnintsiou@gmail.com

Elisavet Grigoriou
eBOS Technologies
Nicosia, Cyprus
elisavetg@ebos.com.cy

Paris Alexandros Karypidis
Department of Economics
Democritus University of Thrace
69100 Komotini, Greece
pakarypi@econ.duth.gr

Theocharis Saoulidis
Sidroco Holdings Ltd
Nicosia, Cyprus
hsaoulidis@sidroco.com

Eleftherios Fountoukidis
Sidroco Holdings Ltd
Nicosia, Cyprus
efountoukidis@sidroco.com

Panagiotis Sarigiannidis
Department of Electrical and Computer Engineering,
University of Western Macedonia
Kozani, Greece
psarigiannidis@uowm.gr,
https://orcid.org/0000-0001-6042-0355

***Abstract*—Digital Twins have been deployed for multiple purposes in Cybersecurity. Cyber-Physical Systems and similar systems are benefited from teaming with this technology. The goal of protecting the main systems or devices is easily reached as Digital Twins are known for their success and flexibility in multitasking and can provide adaptability to any device they clone. honeypots are excellent filtering and protection cyber tools capable of investigating and logging malicious activity around a network or device. This cyber tool mimics any network device or system and lures the attacker into a protected surveillance environment. It is not yet given the freedom, though, to act independently other than managing a few tasks and necessitates human intervention to change tactics or configuration. The framework proposed in this paper combines these two technologies, Digital Twins and honeypot, to fill this gap. The DiTwinIHon framework makes the physical honeypot easily adaptable in any network situation while enhancing its Threat Intelligence and providing additional features for detecting and investigating various threats, such as Advanced Persistent Threats.**

Keywords—Cybersecurity, Digital Twin, honeypot , cyber-attacks, Digital clone, Threat Intelligence

I. INTRODUCTION

During the last five years, many high-profile cyber-attacks on companies, government agencies, websites, institutions, and universities have been reported and recorded. These attacks are making online services and digital systems vulnerable. Thus, more than ever, the field of cybersecurity requires new solutions and tools, such as honeypots, to protect confidential data, systems and devices more effectively. However, it has been observed that using honeypot s, although functional and more modern than other methods, is not a solution by itself, as attack strategies are constantly updated and modified according to the occasion, e.g. smart farming [1]. Meanwhile, developing a prevention and response strategy is slowed by the time-consuming process of analyzing attack data. This time lost in analyzing the first attack should usually be spent on preventing a second one. Using a combination of protection mechanisms seems to be the best practice in such attacks. An excellent answer to this problem is using Digital Twins (DTs) technology on honeypots to improve their threat intelligence. Considering the above, in this paper, a **framework, called DiTwinIHon, for a digitally twinned honeypot is proposed to visualize that thought**.

The DT honeypot framework leverages an existing honeypot with additional features that render it capable of protecting a network or device without the surveillance of humans being necessary. In this framework, the honeypot is given the autonomy to act independently. The deployment of a DT of the honeypot makes this possible. The DT collects real-time data captured from the honeypot and records its states. Simulation, testing and monitoring are facilitated by retrieving and reproducing these states and combining them with the corresponding data. After that, the data produced by the previous procedures are analyzed. A Threat Intelligence module assesses the analyses' results and identifies the necessary changes on the honeypot that are considered appropriate for the situation. Though not demanded, the human factor is not neglected in this framework. Operators can send commands to the DT to reproduce, simulate and test various attack scenarios with real-time considering historical data and honeypot states in a secluded area and visualize the results of these actions.

The remainder of this paper is organized in the following structure. Section 2 presents background on DT technology, cybersecurity threats and previous detection and prevention measures. Section 3 shows related works of similar frameworks that deploy DTs. In section 4, our DT honeypot framework is proposed. Section 5 includes suggested tools which will assist in realizing the DT Representation and Computational models for the proposed framework, and section 6 concludes this paper and presents ideas for the future development of the framework.

II. BACKGROUND

A. Introduction to Digital Twins technology

This section will present the fundamentals and theoretical background of DTs. Secondly, the components that compose the DT and types of DTs will be mentioned to get the gist of what a DT is and how this technology correlates or differentiates itself from other known technologies. Many definitions have been introduced for DT technology: from Grieves and Vickers [2] to Bochert and Rosen [3] and from Abramovici et al. [4] and Schroder et al. [5] to Gabor et al. [6] and Rosen et al. [7]. Since then, many authors have given definitions and descriptions similar to the previous ones. A most recent paper seems to be collecting all of the information of the past attempts into a single definition: "*A Digital Twin is a set of virtual information constructs that*

DOI: 10.1109/CSR57506.2023.10224997

mimics the structure, context, and behaviour of an individual/unique physical asset, is dynamically updated with data from its physical twin throughout its lifecycle, and informs decisions that realize value [8]." This last definition is declarative of the three main components that constitute a DT, according to Grieves [9]: (1) the physical product, (2) the virtual/digital part, and (3) the connection between the physical and virtual parts. The **physical product** refers either to the product itself or its lifecycle. The **virtual part** implies the virtual counterpart that represents the physical one, and the third part refers to a **bidirectional connection** which serves as a way of data transferring between them. Those components enable the DT to identify the traits, copy and synchronize with the physical counterpart. DTs can be categorized based on three criteria: a) the **production** phases, b) the level of **integration**, and c) the **focus** on each task.

According to Grieves [10], based on the ***production phases***, there are **Digital Twin Prototypes (DTP), Digital Twin Instances (DTI) and Digital Twin Aggregate (DTA).** A DTP constitutes a representation model of the physical object that will be manufactured, thus containing a description and an information model. A DTI is a particular physical instance of an object. It could include a list for enumerating individual object parts used in producing this specific instance of an object and every process step followed during its production. A DTA is an accumulation of multiple DTIs which enables querying information about a group of objects [11].

The DT had precursors before reaching its final form, and because of them, there was confusion about their real identity. Those ***precursors are the types of DTs based on the level of integration***: a) **Digital Model**: a digital representation of a planned or existing physical entity where there is no connection between the digital and the physical entities or data transfer between them; b) **Digital Shadow**: a digital representation of a physical entity with a one-way flow of data from the physical to the digital entity. The DT is a digital representation of a physical entity with a *bidirectional connection* between the digital and the physical entities [12].

While the DTs are connected to their physical counterpart, they are helpful for monitoring the actual state of the object, predicting future conditions and states, and remotely improving its condition. DTs were initially deployed in various industrial manufacture phases, namely the *design*, the *operational* and the *disposal* phases [13]. According to the focus on each task, DTs can be of certain types [14]:

DT type	Description
Imaginary DT	A conceptual entity that portrays a non-existent object contains all the information necessary to realize it, including 3D models and specifications about materials and resources.
Monitoring DT	Constitutes a virtual representation of the behaviours and state of an existing physical object.
Predictive DT	Estimates and computes an object's future states and behavioural characteristics with the assistance of predictive analytics based on real-time data acquired by it.
Prescriptive DT	An intelligent digital object that adds intelligence for recommendations and prevention measures based on optimization algorithms and expert heuristics.
Autonomous DT	Takes complete control over the behaviours of the physical object and operates autonomously without human intervention.
Recollection DT	A memory hub that preserves the complete history of a physical object

Table 1*. DT types according to the focus on the individual tasks*

According to their degree of autonomy, DTs can also be categorized as *autonomous*, *partly autonomous* and *not autonomous*. That means a DT of the first category is set to act ultimately and decide based on its own 'intellect'. In the second category, a DT is observed by a human while working, giving results and improving itself and is assisted in the decision-making part, whereas in the third category, it needs strictly human approval of each step of the processes and decisions to be made. DTs can be further employed for protecting critical infrastructure by duplicating the device or network that is prone to attacks which provides additional information for the security experts to conduct: (1) Detection, (2) Investigation and (3) Threat Prevention. The above actions can be implemented on the DT before an attack. This helps the security operators to gain time and give a complete solution to the security leaks they find in their research.

To better understand why DT technology is radically being applied in Cybersecurity, one has to delve deeper into cybersecurity experts' challenges and acknowledge the available tools.

B. Cybersecurity: threats, detection and prevention

Cybersecurity challenges and pending threats such as malware, Distributed Denial of Service (DDoS), Man In The Middle (MITM) attacks and phishing have been raging due to the lack of information for cybersecurity experts to study on and address them adequately. There is usually a combination of multiple threats during a single attack which means that cybersecurity experts need actual strategies to face them effectively. Firewalls and monitoring tools are the first consideration in controlling and observing a system or network.

Cyber-Physical Systems (CPS): Deploying CPSs for large networks and systems assists in managing and observing them as a whole. According to [15], these systems are "*integrations of computation and physical processes*." Such systems "*control physical processes embedded computers and networks*". They have different layers, and human interaction with the system is possible through the device layer, where information about the system is found and decision-making is done. However, these systems lack detection and prevention mechanisms to protect them. Integrating computational and physical components in a network was a step forward in automating real-world tasks and having complete control over a system or network.

Intrusion Detection and Prevention Systems (IDPSs): Among the monitoring tools, IDPSs are considered more effective as they combine system monitoring, analysis and logging of system activity. They can identify possible threats and log information about them, creating reports for cybersecurity experts while taking preventive measures to try and stop or avoid them simultaneously. IDPSs consist of three components to implement the above processing methodology [16] (i) **Sensors-agents**: They are network modules that are responsible for the collection and preprocessing of information in a system, (ii) **Analysis Engine:** The core component of an Intrusion Detection System (IDS), a module

DOI: 10.1109/CSR57506.2023.10224997

that analyzes and classifies information and data, processes them and detects an incident of attack or malicious behaviour, and (iii) **Response Module**: The component that gets the information outcome from the Analysis engine and notifies the security experts. This module can perform limited actions, like activations of rules in a firewall, to diminish an attack and prevent intrusions of this kind on the system.

honeypot s: Another defence mechanism defined as an "*information system resource whose value lies in unauthorized or illicit use of that resource*" [17] is the honeypot . honeypots can mimic the behaviours and responses of a real system or device and trap attackers to block their way to the desired destination. Their main goal is to deceive them and put them into a virtual surveillance box from which intelligence about their activities and methods is collected and recorded. The infrastructure of a honeypot, though, does not only rely on a virtual environment but also consists of real devices to seem more realistic to the attacker's eye, and thus they are categorized as physical or virtual honeypots. Their difference lies in the higher responsiveness that the virtual honeypots offer and the reliability the physical ones show, as they are tough to distinguish from a real target due to their physical existence in the network. They can also be classified as **production and research honeypots** [18] based on the different purposes they serve and as **pure, low-interaction, medium-interaction and high-interaction** based on the level of their interaction with the attacker [19], [20]. Lastly, according to their placement inside a network, they can be defined as a **server, client and hybrid** honeypots [21],[22].

C. Related-Work

Industry 4.0 recommends a standard Layer Model to guide the design steps of a DT. According to [23], **three conditions have to be met for a technically sound DT design** (i) modelling of assets, (ii) decision-making methods and predictive analytics (to support decision-making) and (iii) a knowledge base that is centred around its lifecycle and informs itself with historical and real-time sensor data and relative external information from other databases on the internet.

Frameworks referring to CPSs provide knowledge about the DT capabilities as a Cybersecurity mechanism. In [24] proposed a framework for a Cyber-Physical Digital Twin (CPDT) that aims to provide manufacturing systems with simulation, predictive ability and intelligence for accurate analysis and decision-making. In their paper, they design a CPDT in three layers: the operation, the visualization and the intelligence. The CPDT was capable of: (i)collecting information for the physical assets that were located on the same production floor, (ii) visualizing all the processes and physical assets in real-time by modelling them and synchronizing them with them, and (iii)conducting real-time analysis of historical and current real-time data, which lead to accurate and on-time decision-making.

In [25], the authors show a scenario of applying a DT to the production process for a distributed manufacturing system. In this scenario, when activated, a DT is applied to interoperate with the factory operation schedule. It analyzes the current and past situations by requesting the related information from the sensors/middleware and the database accordingly. The analysis output can be used for managing the production status by receiving manufacturing element data and historical data, assisting in creating a DT model. When activating the DT, the human operators can monitor and track current and historical status accordingly. The DT also assists in decision-making and future production planning as it can simultaneously analyze current and historical data while accessing past production schedules. In [26] proposed a framework for designing a DT for a CPS called CPS Twinning, which emphasizes security assessments for the CPS. Their objective was to provide cyber defence capabilities for CPS operations staff. The framework consists of two components of significant value: a generator that is tasked to automatically create a replica of a physical object and/or network topology and virtual space which can provide two different operation modes: replication and simulation. The framework proposed in [27] paper focuses on security management through various DT processes for ICSs. CPS specification data through sensors and event logs are collected into a Specification and a Historical/State database accordingly. The Specification database includes enough information to create the DT model of the CPS, including security and safety rules predefined in the CPS specifications. The DT can perform basic tasks such as Emulation, Aggregation and Querying of data from the Historical/State database and Monitoring the CPS. Thus, the DT has core abilities to support security operations such as historical data analytics and optimization, simulation and replication. Additional features to the CPS Twinning framework were added by the same authors in a follow-up paper [28], which resulted in a newer framework which shows that it is essential for a cyber DT to keep up with any latest changes and provide a virtual depiction of the physical devices. For the DT to replicate the condition and state of the physical devices, it collects data passively from the Physical Environment. In this way, monitoring of the DT and intrusion detection can be achieved, and at the same time, it makes it easy to spot any difference or deviation between its behaviour and the behaviour of the physical Environment. This feature allows operators to inspect the DTs but restricts the framework to presenting only present states and behaviours, which is solved by providing another feature called "record-and-replay". It can reproduce DT states on demand by storing stimuli produced during a DT state's creation process and assists in analyzing various historical states. The feature of "visualization", which has also been added to the framework, uses the information available from the two features, Monitoring and intrusion detection, and delivers visual "security-relevant" information.

III. A Framework For The Digital Twin Honeypot

The proposed framework shown in **Figure 1** will focus on DT Cybersecurity multitasking around a honeypot while having an ultimate goal: optimization of the honeypot. The initial thought is to build a DT framework to support a honey-pot during its runtime by replicating its features and functionality. The DT will be able to investigate the honeypot's behaviour, performance and operation and provide a simulation area for testing, along with the visual output for the security operators simultaneously. Specific criteria for the DT need to be stated and considered to create

DOI: 10.1109/CSR57506.2023.10224997

the framework. Namely, (a) the DT must be constantly connected to the honeypot with its assets and features to copy, (b) their connection is directed at both parties, which assists the DT in improving itself and the honeypot. The DT initially, (c) needs the acquisition of a plethora of data related to the system's assets to be twinned, the honeypot in this case, and (d) data is passed from the system to the DT through sensors, datasets and external databases that might contain information relevant to the system that can help maintain or improve it. Data is also passed vice-versa, from the DT to the system for commands, Monitoring and optimization. The honeypot itself has the following prebuilt features: (i) Investigations processes, (ii) Lists, and (iii) the DT honeypot will have an additional feature to the prebuilt ones, an intelligent response to automatic actions based on observed activities.

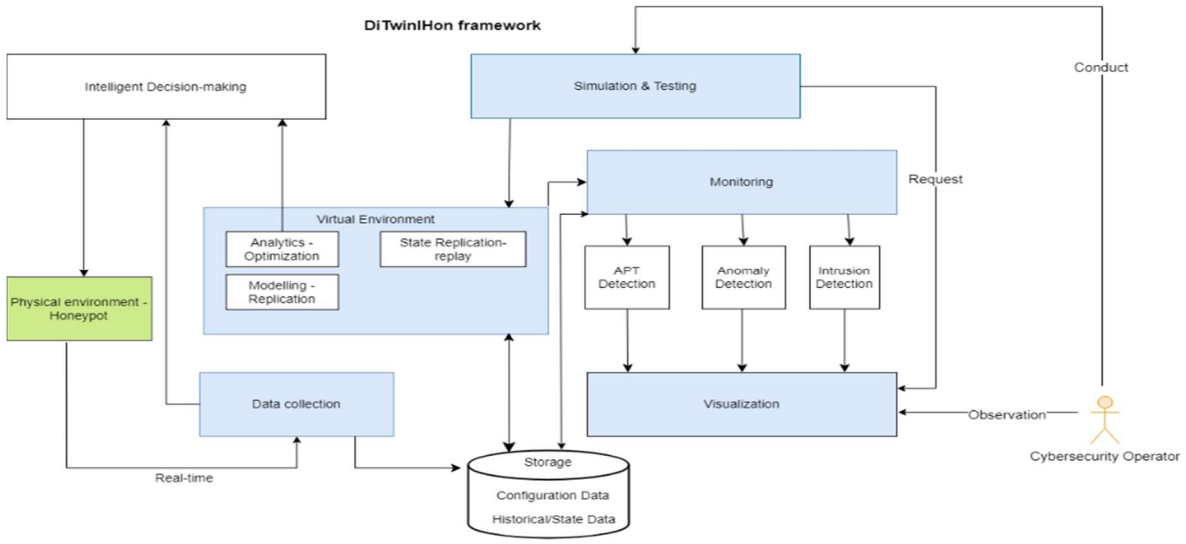

Figure 1. *DiTwinHon framework*

Investigations processes enable analysts to review and comprehend data collected from the honeypot. The honeypot's lists are updated based on the information of the investigation processes. As part of the DT honeypot development cycle, a collection of computational data and representation models will be applied to the honeypot to acquire "intelligence". The DT will perform analytics and processing during the honeypot lifecycle phases, and by using AI algorithms, inferred data acquired during the run time will be incorporated into the DT knowledge base. The DT honeypot will then support customized analytics and will perform a series of cybersecurity tasks. This would result in the honeypot having an intelligent response to automatic actions based on observed activities. There are specific design steps for the DT framework to be created. Digitally twinning the honeypot requires data, so a model needs to be created with the data that describe the real honeypot.

Therefore, **Data Collection and Storage** of the data in a Storage point (a database) for later use is required. After that, the DT acquires the data from the Storage point, and processing of the data begins. The next step is moving to a Modelling-Replication phase, where a digital clone of the real honeypot is being created. Analytics-Optimization and State replication-replay modules are available when the digital twin honeypot (the clone) is ready to run. These extra features provide the digital twin honeypot with functionalities such as Monitoring and Decision-making. The role of the digital twin honeypot is not stop in replicating the current condition, analyzing state and network traffic data, making decisions and Monitoring. Although its main task is to optimize the real honeypot , it also constitutes a virtual environment where cybersecurity operators can conduct Simulations and Testing without fearing damaging or occupying the real honeypot during its runtime. What's more, after the Monitoring processes are done, it informs the human operators with Visualization of the state it is in and its current configurations. This means that, in the end, cyber-security operators have the power to simulate various states and conditions in the virtual Environment, test different configurations on them and visualize the results along with the outcome of the Monitoring of the network traffic data.

Data collection First and foremost, the DT necessitates data collection for it to have accurate information reference to work with during the replication process. The source of information, in this case, is a real honeypot with its features, configuration data and multiple functions which observes and logs the current network traffic. Configuration, current Virtual Machine (VM) state, network traffic and other data that complete the description of the current condition of the real honeypot is collected. The process of collecting data is not done once and for all. The DT requires real-time data collection initially to replicate the honeypot into its digital clone as accurately as possible. Collecting data in real-time also ensures that after its creation, the DT can keep up with any changes in the configuration of the real honeypot. This way, the DT is constantly 'fed' with new network traffic data for processing. Another source for data collection, which is not done in real-time, is the Decision-making outcome which contains information, such as pcap (network packet capture) files, about the best configuration for the honeypot and state data of the best scenario that is worked out after the analysis and the decision-making processes. Data can be obtained from external sources, like databases of other DTs or cloud databases that contain information that either supports functions or provides assistive knowledge for specific tasks.

DOI: 10.1109/CSR57506.2023.10224997

Storage Utilizing databases, every piece of stored information can be considered, aggregated and used for analytics, computation and comparison. The "Storage" in the DT framework depicts the databases that store configuration and network traffic data and log files. Another type of data is stored in the Historical states of the DT, which are the previous and current states of the DT. After the cybersecurity operator requests the State replication-replay component, and after the recreation of the DT model based on the characteristics requested, there is also storing of the new DT model and the data that constitute the new Model. Lastly, the real twin is not Storage's only source of information and data. A DT can associate and authorize other DTs or data storing points on the cloud, fetch information and store it in the Storage. This data stored could be Machine Learning (ML) parameters and models that match the current DT situation and enrich the knowledge base of the DT when comparisons or decisions are made. In this thesis case, global cybersecurity attack definitions, characteristics and defence methods need to be stored for the DT as a knowledge base for the Advanced Persistent Threat (APT), anomaly and intrusion detection task. It is well known that the virus and malware definition and defence methods database must be constantly updated for the DT system to work with up-to-date information and learn to manage threats with cutting-edge techniques. Intelligent decision-making also benefits from relative external data storage.

Virtual Environment. During the replication phase, the DT adopts the same state and characteristics as the real honeypot based on the collected data it obtains from it. For the digital twin honeypot to be able to provide a safe experimentation space and give the ability to analyze and make changes to the honeypot, a Virtual Environment is needed in which any changes in specifications or damage done during the process will not affect the real honeypot. The Virtual Environment extends the functionality of the DT, including analyses and optimization, which leads to intelligent Decision-making. Its advantage is that it can be used as a secure space for experimentation, where Simulation & Testing on the DT is feasible.

Simulation & Testing Simulation of the current state and testing different configurations on the real honeypot is a complex and dangerous task during its run time. When the honeypot is running, it is not advised to make changes or experiment using malware and attacks. In many cases, though, it is necessary to experiment, especially when changes in configuration and testing on new parameters are needed to optimize the existing honeypot. Cybersecurity operators are either forced to make decisions without being able to simulate and observe the results on the honeypot or obliged to disconnect it from the network for a period to test and make additional changes. These methods are both dangerous and time-consuming because, in either case, the network will probably be in danger or unprotected, the log files kept during that period will be useless, and attempts will be in vain. Simulating and testing in an isolated environment provide a solution to this problem while providing room for experimentation with real-time data, and even the case of destroying it would not cause a problem in the real world.

State Replication-replay When simulating and testing various configurations on the real honeypot, previous states are ignored as they cannot be recovered. During the runtime of the DT honeypot, though, this barrier can be overcome by storing every state stimuli in a database (Storage in Fig.1) and reviving it whenever requested. Precisely, the DT honeypot can reproduce DT states on demand by storing stimuli produced during a DT state's creation process. This means that previous states' stimuli are selected inside a database when a cybersecurity operator requires them. Cybersecurity operators can benefit from this feature as they can have a complete DT state outlook of a specific time or event that happened beforehand, whenever needed, can make analyses on those historical states and thus have a better view of the situation by combining previous knowledge.

Monitoring processes acquire previous configuration and network traffic data to compare with the current data.

Intrusion Detection Having an intrusion detected in real-time is not always the most straightforward task for a honeypot, as viewing log files and attempting to figure out mismatches or misbehaviour could take some time when new fast attack methods are on the rise. This task does not seem that complicated when the honeypot is equipped with previous data and can replicate historical states to compare during the investigation. What the honeypot is doing with this module is applying ML algorithms to detect matching patterns with malicious signatures from an attack signature database. The database contains all known attack signatures and thus can be utilized to find such activity and raise an alert.

Anomaly detection When equipped with current and past configuration data and log files, the DT can detect any anomalies -misbehaviour or sudden unwanted changes in its current state- and can alert the operators and simultaneously provide them with a visual representation of the changes. Comparing the monitored activity and a baseline profile built within the honeypot's training phase and with a specific threshold set makes it easy to find any deviation in the monitored activity and consider it malicious.

APT detection APT is a cyber threat that aims to spy and extract valuable information from its target. Those attacks are intended to obtain confidential information, intercept intelligence sent out by attacked computers, and enable the computers to automatically send related intelligence. While other types of attacks usually make their existence clear and hit the system "once and for all", APT attacks do not reveal or uncover themselves, and they introduce themselves in several stages. APTs detection is a demanding task that cannot be done only by viewing previous and current log information. Detecting APTs may take years of logging misbehaviour, malfunctioning or unrequested changes in specific parameters and values and analyzing this information. Although APT detection is a new and not extensively explored area of research, this paper [29] gives a solution by proposing a framework. This framework generates APT attack data and inputs it into a model. Then the Model is trained with Hidden Markov models to recognize, learn and manage to detect the attack pattern. With this method, APT attack stage detection is possible.

Visualization This framework component uses the information available from previously mentioned features, intrusion detection, anomaly detection, and APT detection, and delivers visual "security-relevant" information. It is essential as it provides the cybersecurity operator with visual changes or misbehaviour that could impact the network or the honeypot. With the support of this module, any unexpected

DOI: 10.1109/CSR57506.2023.10224997

changes in status or possible problematic behaviours are made visible and can more easily be spotted when they can be viewed as charts and architectures instead of simple alerts and warnings.

Intelligent Decision-making. After extensive analyses of historical and current data, monitoring network data, and predictive analyses for future states, the DT aims to update the real twin. Decision-making is an integral DT component that enables DT to fulfil that goal. With this component, the DT honeypot identifies the characteristics and the changes that need to be applied to them and can "make a decision" about the physical honeypot for it to work more efficiently. After "making a decision", the DT creates a configuration file and stores it in the Storage. The DT can apply this decision to the real honeypot by swapping its configuration file with the one created and ordering other appropriate changes in its working environment values.

IV. Tools And Methods To Develop Digital Twin Honeypots

A. Data Model Creation

The first step in creating a DT model is to define how and which components will be utilized, what relationships should be formed between them and what connection measures should be taken into consideration (such as authorization of external applications or devices to access or provide data that is valuable during the various tasks or the processing phases). According to [30], no "one-way street" exists for creating a DT model. It can be created using a DT editor or by instantiating a DT based on already made akin DT models. Another way would be combining DT models or specific parts of them that are required. Analyses of an existing DT model and the DT characteristics could also provide the necessary information to create a new similar DT model. As described in previous sections, DTs require 'standard' and, in some cases, specific components to support their functions and tasks. There are certain core components and related technologies that are required for a DT model creation. There are also interrelations and data transfers between DT components and devices [31] and must be described to create an accurate DT model. Depending on the nature of the problem, the creation of a DT model can be described and presented by providing data representation and computational models, as shown in Figure .

B. Data Computational Models

After managing data collection, Storage, and exchange, the DT must process and analyze the data. Analytics and processing during the lifecycle stages of the DT are highly demanded, especially to support the non-stop improvement they promise. Processing real-time and batch-oriented data collected from sensors is done with the help of computational data models [32]. By querying, aggregating, analyzing and processing real-time and batch-oriented data, DTs can optimize themselves, and by deploying ML and statistics on this data, analytics and decision-making are possible.

C. Data representation Models

To create a DT model, the first tasks required are collecting, exchanging data and searching through data. Data representation models can represent the logical part behind those tasks and the related entities. The [33] mentions that when discussing data representation models, we should include the following types of models, Semantic data models, XML-based models, STEP models, and CAPEX models.

Semantic data models are "high-level, user-oriented" data models designed to assist the user in viewing and interacting with the database [34]. XML-based models can encode documents in human and computer-readable formats [35]. A STEP model (Standard for Exchange of Product data) includes data that describe the components or entities entirely by using a formal specification language [36]. CAPEX (computer-aided engineering exchange) is a "meta-model for the storage and exchange of engineering data models" [33]. Ontology models, according to this paper [37] contain rules created by the "concept definition", can contain "conceptual knowledge of a DT", and thus assist in restricting semantic concepts in domain-specific conceptual relationships and are commonly used for database mapping.

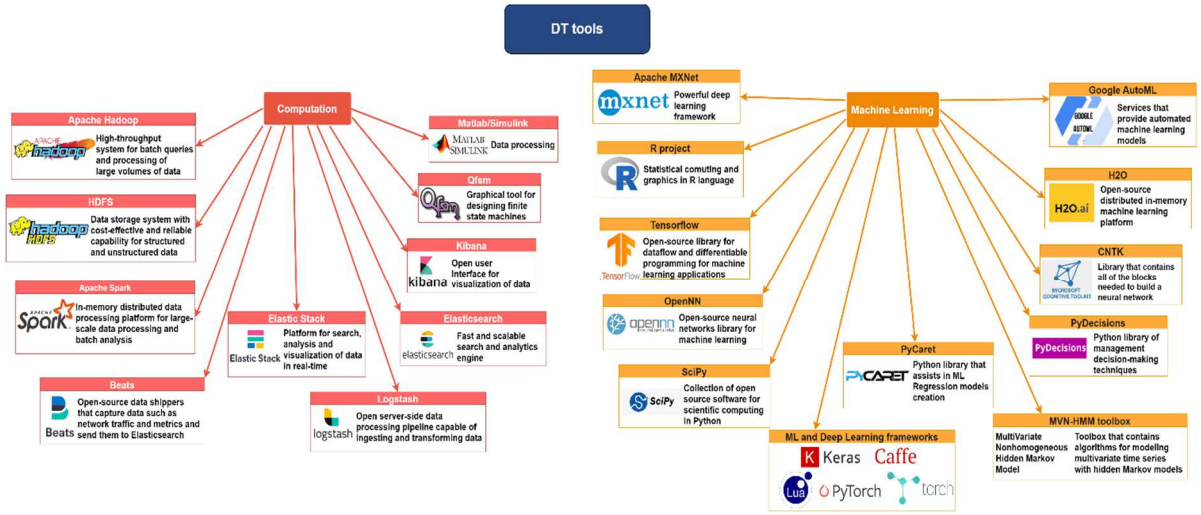

Figure 2. *DT tools: Computation and Machine Learning*

D. Tools to create DT models

To create the DT models, one must choose the relative tools according to the framework. In Figure , easily accessible open-source tools that can be used to implement the following DT tasks are presented:

- **Representation** is supported by semantics, ontology, modelling and XML-based and similar data formats

DOI: 10.1109/CSR57506.2023.10224997

- **Computation** consists of the search, analysis, processing and Visualization of data
- **Communication** between the DT components and the honeypot is done with Machine to Machine (M2M) connectivity and data exchange protocols
- **ML** frameworks, algorithms and platforms support the Simulation, Analytics, Prediction and Decision-making DT modules

send configuration suggestions to the device to work more efficiently and securely.

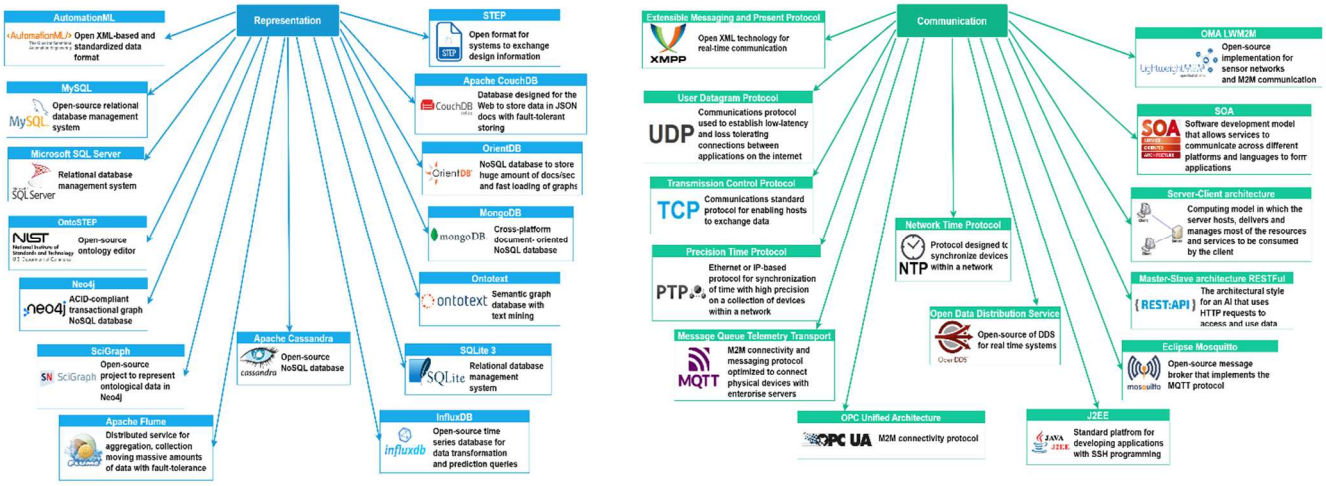

Figure 3. *DT tools: Representation, Communication*

V. CONCLUSION

This paper proposed a framework for a DT honeypot which renders the real honeypot self-updating and self-adaptive in additional network traffic and conditions. This framework is presented as honeypot specific, which means that there are custom functionalities that support the tasks of a real honeypot, and therefore the main focus was its features and the network traffic it logs. The **DiTwinIHon framework** is capable of constantly updating the real honeypot and assisting the human operators in observing both the current and past situations of the network and enabling them to make additional changes when necessary. It provides a wide range of components that enable: 1) data collection, 2) storage, 3) simulation and testing, 4) state replication, 5) monitoring through intrusion, anomaly and APT detection, 6) visualization and 7) intelligent decision-making. Lastly, methods and tools currently available to visualize this framework have been suggested.

Moving on with the framework, extending its capabilities by digitally twinning the Environment of the honeypot would be helpful. Having a generator responsible for extracting the specifications of the physical Environment (such as the network topology or the IoT devices connected) is an initial suggestion. At the same time, the honeypot is a mimicking device and could mimic a CPS or an IoT device. Recreating the Physical Environment gives an advantage to the security operators in such cases as they can observe how specific changes or real attacks on one device (the honeypot) can affect other devices connected to it. It provides a clearer view of the whole network of devices interacting with the honeypot and feedback from those devices. A great addition would be a surveillance component for improving those devices, similar to honeypot decision-making retrieving alerts and logs of their current state and, after analysis, would

ACKNOWLEDGEMENT

This work is a part of the IRIS project. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101021727. This content reflects only the author's view, and the European Commission is not responsible for any use that may be made of the information this publication contains.

REFERENCES

[1] S. Sontowski et al., "Cyber Attacks on Smart Farming Infrastructure," in 2020 IEEE 6th International Conference on Collaboration and Internet Computing (CIC), Atlanta, GA, USA: IEEE, Dec. 2020, pp. 135–143. doi: 10.1109/CIC50333.2020.00025.

[2] M. Grieves and J. Vickers, "Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems," in Transdisciplinary Perspectives on Complex Systems, F.-J. Kahlen, S. Flumerfelt, and A. Alves, Eds., Cham: Springer International Publishing, 2017, pp. 85–113. doi: 10.1007/978-3-319-38756-7_4.

[3] S. Boschert and R. Rosen, "Digital Twin—The Simulation Aspect," in Mechatronic Futures, P. Hehenberger and D. Bradley, Eds., Cham: Springer International Publishing, 2016, pp. 59–74. doi: 10.1007/978-3-319-32156-1_5.

[4] M. Abramovici, J. C. Göbel, and P. Savarino, "Virtual Twins as Integrative Components of Smart Products," in Product Lifecycle Management for Digital Transformation of Industries, R. Harik, L. Rivest, A. Bernard, B. Eynard, and A. Bouras, Eds., in IFIP Advances in Information and Communication Technology, vol. 492. Cham: Springer International Publishing, 2016, pp. 217–226. doi: 10.1007/978-3-319-54660-5_20.

[5] G. N. Schroeder, C. Steinmetz, C. E. Pereira, and D. B. Espindola, "Digital Twin Data Modeling with AutomationML and a Communication Methodology for Data Exchange," IFAC-Pap., vol. 49, no. 30, pp. 12–17, 2016, doi: 10.1016/j.ifacol.2016.11.115.

DOI: 10.1109/CSR57506.2023.10224997

[6] T. Gabor, L. Belzner, M. Kiermeier, M. T. Beck, and A. Neitz, "A Simulation-Based Architecture for Smart Cyber-Physical Systems," in 2016 IEEE International Conference on Autonomic Computing (ICAC), Wuerzburg, Germany: IEEE, Jul. 2016, pp. 374–379. doi: 10.1109/ICAC.2016.29.

[7] R. Rosen, G. von Wichert, G. Lo, and K. D. Bettenhausen, "About The Importance of Autonomy and Digital Twins for the Future of Manufacturing," IFAC-Pap., vol. 48, no. 3, pp. 567–572, 2015, doi: 10.1016/j.ifacol.2015.06.141.

[8] H. Aydemir, U. Zengin, and U. Durak, "The Digital Twin Paradigm for Aircraft Review and Outlook," in AIAA Scitech 2020 Forum, Orlando, FL: American Institute of Aeronautics and Astronautics, Jan. 2020. doi: 10.2514/6.2020-0553.

[9] M. W. Grieves, "Product lifecycle management: the new paradigm for enterprises," Int. J. Prod. Dev., vol. 2, no. 1/2, p. 71, 2005, doi: 10.1504/IJPD.2005.006669.

[10] M. Grieves, "Origins of the Digital Twin Concept," 2016, doi: 10.13140/RG.2.2.26367.61609.

[11] S. V. Nath, P. van Schalkwyk, and D. Isaacs, Building industrial digital twins: design, develop, and deploy digital twins solutions for real-world industries using Azure Digital Twins. Birmingham Mumbai: Packt, 2021.

[12] A. Fuller, Z. Fan, C. Day, and C. Barlow, "Digital Twin: Enabling Technologies, Challenges and Open Research," IEEE Access, vol. 8, pp. 108952–108971, 2020, doi: 10.1109/ACCESS.2020.2998358.

[13] R. Minerva, G. M. Lee, and N. Crespi, "Digital Twin in the IoT Context: A Survey on Technical Features, Scenarios, and Architectural Models," Proc. IEEE, vol. 108, no. 10, pp. 1785–1824, Oct. 2020, doi: 10.1109/JPROC.2020.2998530.

[14] C. Verdouw, B. Tekinerdogan, A. Beulens, and S. Wolfert, "Digital twins in smart farming," Agric. Syst., vol. 189, p. 103046, Apr. 2021, doi: 10.1016/j.agsy.2020.103046.

[15] E. A. Lee, "Cyber-Physical Systems - Are Computing Foundations Adequate?," p. 9.

[16] P. I. Radoglou-Grammatikis and P. G. Sarigiannidis, "Securing the Smart Grid: A Comprehensive Compilation of Intrusion Detection and Prevention Systems," IEEE Access, vol. 7, pp. 46595–46620, 2019, doi: 10.1109/ACCESS.2019.2909807.

[17] L. Spitzner, "honeypots: catching the insider threat," in 19th Annual Computer Security Applications Conference, 2003. Proceedings., Las Vegas, Nevada, USA: IEEE, 2003, pp. 170–179. doi: 10.1109/CSAC.2003.1254322.

[18] L. Spitzner, "The Value of honeypots, Part One: Definitions and Values of honeypots," 2001. [Online]. Available: http://www.securityfocus.com/infocus/1492

[19] M. Hernandez y Lopez and C. Francisco Lerma Reséndez, "honeypots: Basic Concepts Classification and Educational Use as Resources in Information Security Education and Courses," presented at the InSITE 2008: Informing Science + IT Education Conference, 2008. doi: 10.28945/3186.

[20] S. A. Joshi R.C., honeypots: A new paradigm to information security. 2011.

[21] L. Zobal, D. Kolar, and R. Fujdiak, "Current State of honeypots and Deception Strategies in Cybersecurity," in 2019 11th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), Dublin, Ireland: IEEE, Oct. 2019, pp. 1–9. doi: 10.1109/ICUMT48472.2019.8970921.

[22] S. Kumar, R. Sehgal, and J. S. Bhatia, "Hybrid honeypot framework for malware collection and analysis," in 2012 IEEE 7th International Conference on Industrial and Information Systems (ICIIS), Chennai, India: IEEE, Aug. 2012, pp. 1–5. doi: 10.1109/ICIInfS.2012.6304786.

[23] H.-M. Ríos, José Juan ,Oliva, Manuel, Mas, Fernando, "Product Avatar as Digital Counterpart of a Physical Individual Product: Literature Review and Implications in an Aircraft," Jul. 2015. doi: 10.3233/978-1-61499-544-9-657.

[24] W. D. Lin and M. Y. H. Low, "Concept and Implementation of a Cyber-Physical Digital Twin for a SMT Line," p. 5, 2019.

[25] K. T. Park *et al.*, "Design and implementation of a digital twin application for a connected micro smart factory," *Int. J. Comput. Integr. Manuf.*, vol. 32, no. 6, pp. 596–614, Jun. 2019, doi: 10.1080/0951192X.2019.1599439.

[26] 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*, Incheon Republic of Korea: ACM, May 2018, pp. 61–72. doi: 10.1145/3198458.3198464.

[27] M. Dietz and G. Pernul, "Unleashing the Digital Twin's Potential for ICS Security," *IEEE Secur. Priv.*, vol. 18, no. 4, pp. 20–27, Jul. 2020, doi: 10.1109/MSEC.2019.2961650.

[28] M. Eckhart and A. Ekelhart, "A Specification-based State Replication Approach for Digital Twins," in *Proceedings of the 2018 Workshop on Cyber-Physical Systems Security and PrivaCy*, Toronto Canada: ACM, Jan. 2018, pp. 36–47. doi: 10.1145/3264888.3264892.

[29] I. Ghafir *et al.*, "Hidden Markov Models and Alert Correlations for the Prediction of Advanced Persistent Threats," *IEEE Access*, vol. 7, pp. 99508–99520, 2019, doi: 10.1109/ACCESS.2019.2930200.

[30] L. Stojanovic *et al.*, "Methodology and Tools for Digital Twin Management—The FA3ST Approach," *IoT*, vol. 2, no. 4, pp. 717–740, Nov. 2021, doi: 10.3390/iot2040036.

[31] M. Liu, S. Fang, H. Dong, and C. Xu, "Review of digital twin about concepts, technologies, and industrial applications," *J. Manuf. Syst.*, vol. 58, pp. 346–361, Jan. 2021, doi: 10.1016/j.jmsy.2020.06.017.

[32] Q. Qi *et al.*, "Enabling technologies and tools for digital twin," *J. Manuf. Syst.*, vol. 58, pp. 3–21, Jan. 2021, doi: 10.1016/j.jmsy.2019.10.001.

[33] V. Damjanovic-Behrendt and W. Behrendt, "An open source approach to the design and implementation of Digital Twins for Smart Manufacturing," *Int. J. Comput. Integr. Manuf.*, vol. 32, no. 4–5, pp. 366–384, May 2019, doi: 10.1080/0951192X.2019.1599436.

[34] M. Hammer and D. McLeod, "The semantic data model: a modelling mechanism for data base applications," in *Proceedings of the 1978 ACM SIGMOD international conference on management of data - SIGMOD '78*, Austin, Texas: ACM Press, 1978, p. 26. doi: 10.1145/509252.509264.

[35] S. S. Choi, T. H. Yoon, and S. D. Noh, "XML-based neutral file and PLM integrator for PPR information exchange between heterogeneous PLM systems," *Int. J. Comput. Integr. Manuf.*, vol. 23, no. 3, pp. 216–228, Mar. 2010, doi: 10.1080/09511920903443234.

[36] "Procedure for the maintenance of the STEP Module and Resource Library. ISO TC 184/SC 4 N2538, 2009."

[37] S. Singh *et al.*, "Data management for developing digital twin ontology model," *Proc. Inst. Mech. Eng. Part B J. Eng. Manuf.*, vol. 235, no. 14, pp. 2323–2337, Dec. 2021, doi: 10.1177/0954405420978117.

[38] C. Huang, J. Han, X. Zhang, and J. Liu, "Automatic Identification of honeypot Server Using Machine Learning Techniques," *Secur. Commun. Netw.*, vol. 2019, pp. 1–8, Sep. 2019, doi: 10.1155/2019/2627608.