Securing Industrial Systems: A Testbed for Cyber-Defense Evaluation and Data Collection

Raffaele Cuorvo, Nicola d'Ambrosio, Domenico Iorio, Gaetano Perrone, Simon Pietro Romano
Dept. of Electrical Engineering and Information Technology
University of Naples Federico II
Naples, Italy
raf.cuorvo@studenti.unina.it, nicola.dambrosio2@unina.it, domeni.iorio@studenti.unina.it,
gaetano.perrone@unina.it, spromano@unina.it

Abstract—**Over recent years, many Industrial Control System (ICS) components have been exposed to both the Internet and corporate networks to enhance the management of industrial processes. However, this increased exposure has often taken place without adequate consideration for cybersecurity, making industrial networks more vulnerable to cyberattacks. In this context, digital twins have emerged as innovative solutions to evaluate novel cyber-defense strategies that can mitigate threats affecting industrial networks. Unfortunately, to the best of the authors' knowledge, there is no digital twin that is flexible enough to integrate both physical and virtualized components according to user preferences while simultaneously supporting novel approaches based on the Software-Defined Networking (SDN) paradigm. To address these issues, we developed a flexible hybrid/virtual digital twin that mimics a physical Microgrid testbed known as EPIC. Specifically, our solution leverages virtualization and containerization to create a lightweight platform that can include the widest possible range of vulnerabilities. Furthermore, we employ Open vSwitch to implement SDN-based methodologies and integrate physical components into our platform. Lastly, we provide a comprehensive tool that collects all possible logs from the testbed.**

Index Terms—**Industrial Control System, Smartgrid, Testbed, Software Defined Network, Digital Twin**

I. Introduction

Industrial Control Networks consist of several devices, collectively known as *Industrial Control Systems* (ICSs), designed to monitor and control industrial processes across various sectors such as energy, manufacturing, and oil and gas. Historically, these systems employed proprietary standards and protocols developed without any security concerns. Consequently, companies relied on security-through-obscurity to prevent attackers from controlling these systems for malicious purposes. This security measure aims to minimize the attack surface of industrial networks and conceal all possible details about the configuration and implementation of each device. However, with the advent of the *Industrial Internet of Things* (IIoT), several ICS components have been exposed to the Internet as well as corporate networks. This increased connectivity has expanded the attack surface and made ICSs more vulnerable to cyber threats. As a result, more than a security-through-obscurity methodology is required [1]. In this context, it is crucial to bolster the security posture of industrial networks by implementing innovative cyber-defense strategies and conducting red team activities to test their robustness. Nevertheless, assessing these industrial cyber-defense strategies through red team activities cannot be directly performed on the network infrastructure to prevent disruption or malfunctioning of daily industrial operations. To this end, Digital Twins have emerged as an innovative solution for evaluating the effectiveness of cyber-defense strategies in ICS networks [2]. Indeed, these environments replicate the functionality of real-world ICS networks and enable researchers to simulate various attack scenarios and defense mechanisms without risking operational downtime or compromising safety.

Unfortunately, to the best of the authors' knowledge, there are currently no available digital twin platforms that are flexible enough to integrate physical, virtualized, or containerized components according to user preferences while simultaneously supporting the seamless implementation of innovative cyber-defense strategies that rely on the *Software-Defined Networking* (SDN) paradigm. The ability to customize the integration of physical, virtualized, and containerized components is crucial for creating a high-fidelity simulation environment that meets specific user requirements while also minimizing costs. Furthermore, the absence of support for implementing advanced SDN-based cyber-defense strategies, such as *Moving Target Defense* (MTD), significantly hampers the effectiveness of these platforms in evaluating cutting-edge approaches within the context of ICS networks.

To address these issues, we have developed a hybrid/virtual digital twin that mimics a physical Microgrid testbed known as EPIC [3], aimed at validating cutting-edge SDN-based cyber-defense strategies. This platform is highly customizable to meet user preferences in terms of simulation fidelity. Industrial components can be deployed through virtualization (using containers or virtual machines) or as physical implementations, depending on the specific vulnerabilities the user wants to replicate. Containers are the ideal choice for implementing vulnerabilities related to industrial applications or communication protocols between ICS components. Virtual machines are best suited for injecting either kernel-based vulnerabilities or, more in general, vulnerabilities related to Windows-based systems. Physical components are typically preferred for analyzing the security implementation of a specific component developed by a particular vendor. These devices are connected to each other through Open vSwitches [4]. Unlike other methods, Open

978-3-903176-66-9 ©2024 IFIP

vSwitches facilitate the seamless interconnection of containers, virtual machines, and physical devices while enabling the network to be managed through the SDN paradigm.

Furthermore, a data collection tool was developed to gather information from various sources. The collected data not only enhance our understanding of the impact of executed cyberattacks but also serve as a valuable dataset for further research. Specifically, the tool collects raw network data, the commands launched to execute the attack, and the performance metrics of containers and virtual machines (CPU and RAM usage). Subsequently, the collected dataset was uploaded to the ELK stack to enable in-depth data analysis and generate an aggregated visual overview.

To summarize, we have created a digital twin of the EPIC [3] testbed, which offers the following key contributions:

- it provides a flexible implementation framework allowing users to choose between virtual machines, containers, or physical components;
- it provides a network architecture relying on the use of Open vSwitches;
- it introduces a tool suite enabling the comprehensive acquisition of raw network data, attack execution commands, and performance metrics of containers and VMs;
- it makes the source code required to deploy the proposed testbed publicly accessible on GitHub [5].

The remaining sections of the paper are structured as follows. Section III provides a detailed description of the EPIC testbed. Section IV presents an overview of the proposed architecture and underlying approach. Section IV delves into the design of our collection tool suite. Section VI introduces a series of attacks that can be executed against our testbed. Section II provides a review of the related works in the field. Finally, Section VII concludes the paper.

II. Related Works

Testbeds and digital twins offer a controlled environment where researchers and penetration testers can assess the security posture of ICS architectures without disrupting daily industrial operations. Indeed, these controlled environments allow for assessing the effectiveness of security controls, identifying vulnerabilities, simulating attacks, and implementing incident response procedures. In this context, researchers have developed a range of physical, hybrid, and virtual testbeds to underscore the aforementioned tasks [6].

Physical testbeds incorporate industrial and network components to create a realistic scenario that facilitates validating security solutions under conditions that closely resemble operational settings. However, their construction is costly and time-consuming. Notable examples of physical testbeds include the HAI Testbed and EPIC. The HAI Testbed [7] [8] emulates a water treatment system by employing three independent ICSs: a GE turbine, an Emerson boiler, and a FESTO water treatment system. Furthermore, the authors executed several attacks (physics-based attacks, network-based attacks, and system-based attacks) and established a logging system to capture data from Linux and Windows virtual machines. EPIC [3] emulates a real-world electric power system, employing PLCs and IEDs that communicate via the IEC 61850 protocol. These devices are distributed across four different subnets: Generation, Micro-grid, Transmission, and Smart Home, each with specific responsibilities within the overall system. Furthermore, the authors conducted false data injection, physical damage, and malware attacks to extract a dataset that can be used for subsequent research.

Virtual testbeds, on the other hand, rely entirely on software-based simulations and virtualized components. These testbeds offer a scalable and cost-effective solution for testing and validating security measures without physical hardware. Nevertheless, the simulations of these environments provide a lower degree of fidelity than physical testbeds. EPIC TWIN and VICSORT are two notable virtual testbeds in this category. EPIC TWIN [9], which serves as the digital twin of the aforementioned EPIC testbed, simulates the physical processes using Simulink and virtualizes industrial devices mostly through virtual machines. The communication protocols employed between these components include MQTT, MMS, and GOOSE. Similarly, our work emulates EPIC, but it enables the coexistence of both virtual and physical devices while also integrating the use of the SDN paradigm. VICSORT [10] aims to provide a more resource-efficient version of the water supply testbed GRFICS [11]. It specifically leverages LXD containers and Kernel-based Virtual Machines (KVM) to enhance portability and facilitate deployment in cloud environments.

Hybrid testbeds combine elements of both physical and virtual testbeds, offering a balanced approach that leverages the strengths of each. These testbeds can integrate real hardware with virtual components. For instance, authors in [12] presented a testbed that reproduces the behavior of a water distribution system and consists of seven tanks at the physical layer controlled by boards. The communication protocol utilized in the testbed is Modbus, while the implemented attack scenarios focus on data manipulation. Additionally, authors in [13] proposed a low-cost hybrid testbed designed for security education purposes. This testbed employs Raspberry Pi units to implement a SCADA system, an OpenPLC, and an HMI, with the physical process involving several step motors. Moreover, the authors used Modbus TCP as the communication protocol and provided various attack scenarios, such as Denial of Service and Man-in-the-Middle (MiTM) attacks. Similarly, another open-source hybrid testbed, KIPPO4INDUSTRY [14], was developed to train students to address threats in OT networks. This testbed utilizes Raspberry Pi units to simulate PLCs, and to simplify deployment, each PLC is installed in a Docker container. Lastly, authors in [15] developed a hybrid testbed to collect raw network data for testing new cyber-physical security monitoring and detection technologies. Within this testbed, only the network components are physical, while each industrial device is emulated within a virtual machine. The testbed supports a variety of attack simulations, including, but not limited to, denial of service (DoS), MiTM attacks, and malware infiltration. However, it is important to note that raw network data alone may be insufficient for building a comprehensive dataset. Moreover, the absence of physical industrial devices in the testbed limits the ability to integrate vulnerabilities that can not be emulated in a virtualized environment.

Our solution distinguishes itself by leveraging the strengths of both physical and virtual testbeds while addressing their limitations. Through a highly automated and modular approach to network configuration and virtualization, our testbed allows users to seamlessly substitute physical components with their virtual counterparts. A key innovation that further differentiates our solution is the integration of SDN capabilities. Unlike traditional testbeds, our SDN-enabled digital twin enables the evaluation of innovative cyber-defense approaches that rely on the SDN paradigm.

III. EPIC Testbed Layout

This section provides a comprehensive analysis of the EPIC [3] testbed which replicates an Alternating Current Microgrid. Understanding the structure of EPIC is crucial for determining which elements must be replicated in our digital twin to achieve the aforementioned objectives and which components may be excluded. A diagram detailing the electric layout of the EPIC testbed is depicted in Figure 1.

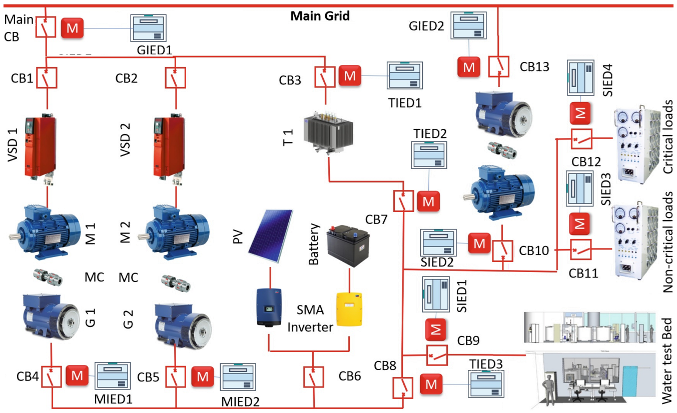


Fig. 1: EPIC electric layout [3]

The EPIC testbed is structured in four different parts, each having a specific responsibility. The Generation Zone is accountable for providing electricity to the testbed from the Main Grid. The Microgrid Zone contains all components involved in generating electrical power within the testbed, such as motor-driven generators. The Transmission Zone isolates the generation components from the load. The Smart Home Zone encompasses all the electrical loads of the testbed. All electrical components within these zones are controlled by circuit breakers (labeled as CB-X in Figure 1). Some of these circuit breakers are managed by specific Intelligent Electronic Devices (IEDs), identified as GIED-X, MIED-X, TIED-X, or SIED-X in Figure 1, while the remaining circuit breakers are automatically opened when unsafe conditions occur. These IEDs operate under a hierarchical control structure where each zone is managed by a dedicated Programmable Logic Controller (PLC). The PLCs for each zone are overseen by a General PLC, which the operator controls via a Supervisory Control and Data Acquisition (SCADA) system. The hierarchical control structure is illustrated in Figure 2.

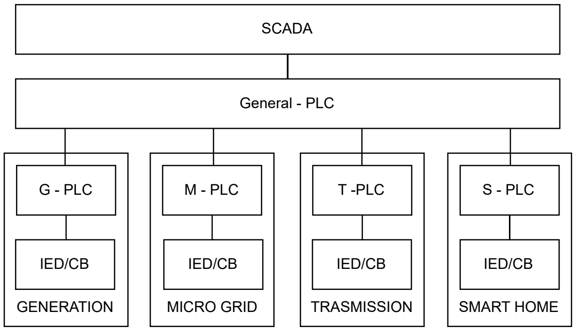


Fig. 2: Hierarchical Control Structure

Based on the EPIC description, we replicate all components involved in managing the control structure outlined above, along with their associated communications, in order to assess novel cyber-defense mechanisms and collect data for further research. This replication specifically involves emulating the SCADA system, as well as the PLCs, IEDs, and managed CBs. Notably, we decided not to simulate the physical process because it does not expand the attack surface. Our focus is on the industrial application and communication layers where most cyber vulnerabilities are likely to be exploited. In future work, we plan to also integrate the physical process simulation, to better understand the consequences of cyber intrusions.

IV. Testbed Architecture

This section provides a detailed description of our testbed, organized into four subsections. The first Subsection gives an overview of the testbed. The second Subsection introduces the network model and communication protocols. The third Subsection discusses the physical and simulated industrial components. The fourth Subsection delves into the configuration of all virtual and physical network devices.

A. Testbed Overview

Our Microgrid architecture is created using a hybrid testbed approach that combines physical and virtual components. This approach allows us to design a product with features from both physical and virtual testbeds. Indeed, we can easily edit scenarios with virtual devices while ensuring strict simulation adherence to reality. Furthermore, every physical component in the testbed is also replicated virtually. This dual capability ensures that the testbed can be utilized effectively even in the absence of hardware components. Details on how to replace these components will be provided in Subsection IV-D.

Figure 3 depicts the physical network architecture of the proposed testbed, which comprises four components: an Industrial Switch[1], four ABB PLCs[2], a Raspberry PI4[3], and a

[1] Perle IDS-409-1SFP Industrial Managed Switch

[2] ABB PM554-TP-ETH PLC

[3] Raspberry PI4 equipped with 4 GB RAM

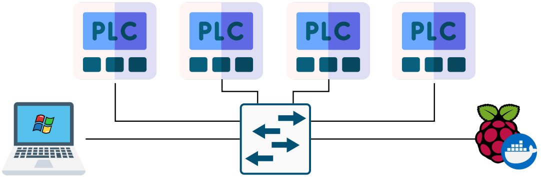


Fig. 3: Physical network architecture

standard HP laptop[4]. The Industrial Switch enables network communication among these components. It is worth noting that the switch must support the 802.1Q standard as the network segregation relies on VLANs for implementation. The four ABB PLCs replicate the control logic of the G-PLC, M-PLC, T-PLC, and S-PLC discussed in Section III. The Raspberry Pi and the HP laptop are utilized to deploy all virtual components of the testbed.

To deploy virtual testbed components, we utilize two different virtualization techniques. The HP notebook uses a Virtualbox Hypervisor to run virtual machines (VMs), while the Raspberry Pi relies on the Docker engine to run containers. We use both technologies to benefit from their strengths and overcome limitations. Container-based virtualization uses fewer resources than traditional virtual machines, but it can have limitations in reproducing specific environments, such as the Windows system. Moreover, our testbed includes four physical ABB PLCs, which play a crucial role in incorporating behaviors and vulnerabilities that virtual components cannot replicate in our attack scenarios.

B. Network model

The network model of our scenario is described here in more detail. Figure 4 displays the network design of our testbed. Micro Grid LAN is segmented into six subnets: (*i*) SCADA, (*ii*) Control, (*iii*) Transmission, (*iv*) Micro Grid, (*v*) Smart Home, and (*vi*) Generation.

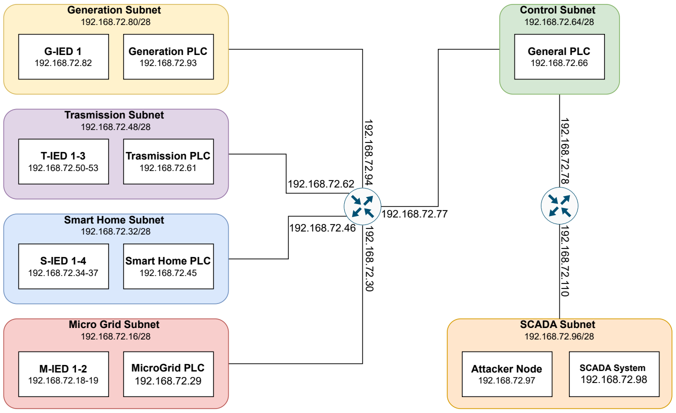


Fig. 4: High-level testbed network model

The SCADA Subnet hosts a SCADA workstation employed by operators to monitor and manage the status of controlled Circuit Breakers. The Control Subnet includes the General PLC, which receives commands from the SCADA workstation and coordinates other PLCs. The other subnets have IEDs controlled by dedicated PLCs. In order to enable communication between different subnets, we have installed two routers, namely, the Main Gateway and the Control Gateway. The Control Gateway links the SCADA and Control networks, while the Main Gateway connects the other subnets. These network devices exclusively rely on directly connected routes to fill their routing tables. Consequently, the SCADA subnet is unable to communicate with other subnets, with the exception of the Control subnet, and vice versa.

It is important to mention that Modbus is the only industrial communication protocol used in our testbed. This protocol is widely used in the industry and allows devices to exchange control messages with each other. However, it was created many years ago with little emphasis on security. As a result, it lacks features that ensure confidentiality, integrity, and availability in communication. Indeed, data exchanged between devices is not encrypted, which makes it simple for attackers to intercept, read, and even modify messages. Nonetheless, the flexibility offered by the testbed architecture enables the community to substitute components in order to employ other industrial protocols.

C. Testbed Components

In this section, we will give a thorough explanation of all the physical and virtual components used to recreate the previously mentioned environment and shown in Fig. 4.

- **SCADA Workstation.** A Windows 7 Virtual Machine is used to implement this node, featuring a web-based HMI that enables human operators to control industrial equipment. The HMI is created using Node-RED, a flow-based programming tool commonly utilized for IoT applications and task automation. Additionally, the server runs an SMB server that is vulnerable to EternalBlue.
- **General-PLC.** A headless container simulates the General PLC, running a Modbus Server and Web-HMI. The PyModbus library powers the Modbus/TCP server, sending control messages to other PLCs. The Web-HMI enables operators to monitor the status of PLCs via CGI-generated web pages. Notably, the web server is vulnerable to the ShellShock exploit.
- **Physical PLCs.** Four ABB PLCs are used to emulate the S-PLC, M-PLC, T-PLC, and G-PLC, respectively. These devices receive commands from the General PLC and control the corresponding IED devices.
- **Intelligent Electronic Devices (IEDs).** Each of the four zones contains a different number of IEDs. Specifically, the Smart Home subnet has four devices, the Transmission subnet has three, the Micro Grid subnet has two, and the Generation subnet has one. These devices are implemented similarly to General-PLC but do not expose any web service.
- **Gateways.** The Main Gateway and Control Gateway are implemented using Open vSwitches. These components

[4] HP laptop equipped with i5-1135G7 processor and 8 GB RAM

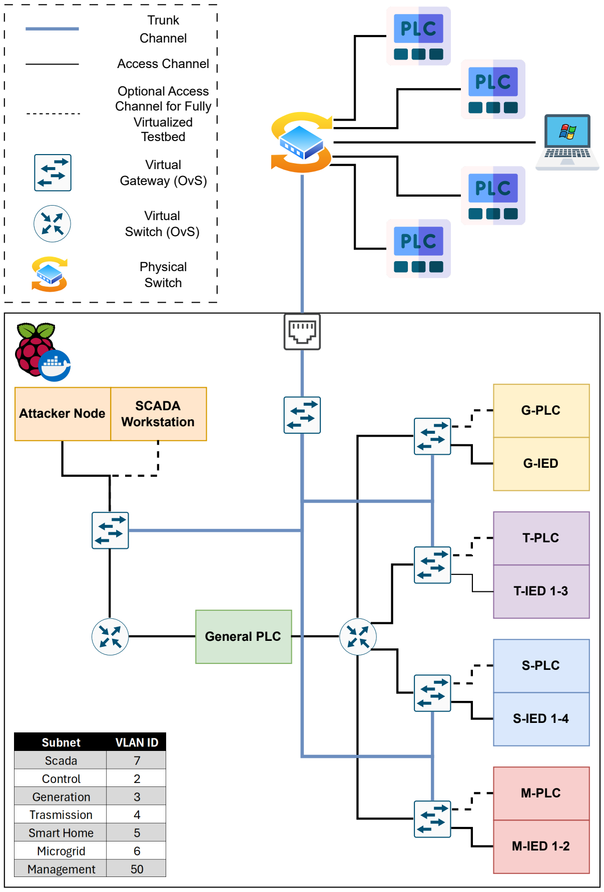
are managed by an SDN Controller, which is deployed within a container that is completely isolated from the testbed network. Additionally, these nodes expose an SSH server with weak password authentication (*admin:admin*).

- **Attack launcher.** A container designed to execute scripts for exploiting vulnerabilities within our testbed.

Notably, SCADA Workstation and physical PLCs can also be deployed as containers. This configuration lets the testbed run on a commercial laptop with minimal resources and eliminates the need for hard-to-find physical components. However, deploying the platform as a containerized virtual testbed may limit its capabilities. Containers do not support the injection of kernel-based vulnerabilities or the emulation of Windows systems within the platform. Physical components are essential for evaluating the security of specific implementations of proprietary standards, which are often employed in industrial contexts. Therefore, users must carefully consider their objectives and select the configuration that best aligns with their requirements.

D. Network Implementation

This section delves into the network configuration of Open vSwitches (OvS) utilized to enable communication between all virtual and physical components. The precise configuration utilized in our testbed is illustrated in Figure 5.



Subnet	VLAN ID
Scada	7
Control	2
Generation	3
Trasmission	4
Smart Home	5
Microgrid	6
Management	50

Fig. 5: Low-Level testbed network architecture

The virtual network is entirely managed using OvS switches, which can be logically divided into two categories: Virtual Switches and Virtual Gateways. Virtual Switches enable communication between devices within the same subnet, whether they are containers or VMs, while also ensuring network segmentation through VLANs. On the other hand, Virtual Gateways are responsible for routing traffic between different subnets. The configuration of these switches is accomplished in three stages. The first stage involves the deployment of the switches as defined within our testbed network architecture. In the second stage, all necessary patch ports and Virtual Ethernet (veth) interfaces are created to construct the infrastructure depicted in Figure 5. Patch ports are utilized to connect different OvS instances, while veth interfaces are employed to attach containers or VMs to the Open vSwitches. These ports can be configured in either trunk or access mode, depending on specific network requirements. Specifically, ports connecting two Open vSwitches or linking an Open vSwitch with the physical Ethernet port of the host system are configured in trunk mode. In contrast, ports used for other connections are set to access mode and assigned VLAN IDs[5] according to the convention shown in Figure 5. The final stage involves linking the ports created in the previous stage according to the network architecture depicted in Figure 5. This ensures that all components are correctly interconnected and that the network operates as designed. It is also crucial that the physical switch adheres to this VLAN convention to maintain consistency across the network. Furthermore, using the methodology illustrated above, users can substitute physical components in the testbed with their virtual counterparts by integrating the corresponding virtual versions.

V. Data Collection

This section explores the suite of data collection tools utilized to gather valuable information from the testbed. The objective is to obtain an extensive range of data essential for evaluating the system under test and to provide a dataset suitable for subsequent research endeavors. The data collection methodologies employed include:

- **Raw network traffic:** packets for both inbound and outbound traffic across all virtual network interfaces within the testbed are captured using the *tcpdump* utility. The collected traffic data is then converted into JSON format to facilitate efficient storage, retrieval, and subsequent analysis. This entire process, from collection to conversion, is performed using the following command:

```
tshark -r "$pcap_file" -T ek | jq -c
↪  'del(.index._type)' >
↪  "pcap_json/${filename}.json"
```

- **Docker Container Performance:** CPU and RAM usage statistics for each container deployed on the platform are gathered using the *docker stats* utility.

[5] The Management VLAN hosts the SDN controller responsible for managing all Open vSwitches within the architecture. The Management VLAN is omitted in Figure 5 to enhance the clarity of the diagram.

- **PLC and IED Status:** the status of each virtual PLC and IED is stored in a file using a purpose-built utility.
- **Virtual Machine Logs:** relevant data from VMs are collected using lightweight data shippers known as Beats, such as *Winlogbeat* for Windows-based VMs. These tools capture system and event logs, transmitting them in real-time to a centralized storage.
- **Commands Executed:** the collection of all commands executed within the attacker node is facilitated by a tool known as CyberRange-Collector [16], which is deployed directly on the attacker node.

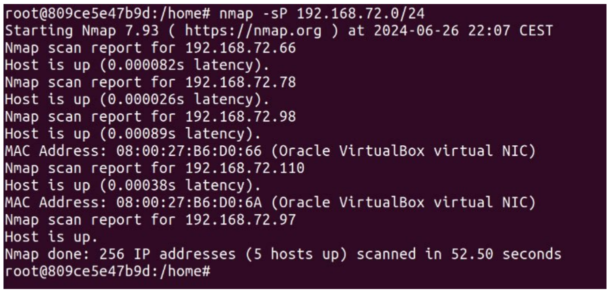
After collection, the data are uploaded to the ELK stack to enable comprehensive data analysis and visualization.

VI. Attacks & Defense

This section aims to showcase the utilization of the testbed for simulating real-world industrial attacks while also implementing SDN-based cyber-defense approaches to mitigate these threats effectively. To this end, an ARP poisoning attack was executed in order to alter Modbus messages exchanged between the SCADA system and the General PLC (Subsection VI-A). Furthermore, we employed an MTD Redundancy strategy within our testbed to divert the aforementioned malicious attack in a controlled environment (Subsection VI-B).

A. On-the-Fly Packet Modification

This attack aims to alter the packets exchanged between the SCADA Workstation and the Master PLC. It is carried out through a series of malicious actions. The initial phase involves a reconnaissance attack in which network scanning techniques are employed to identify all hosts accessible from our node. Specifically, a ping scan is conducted to identify reachable devices within the digital twin.



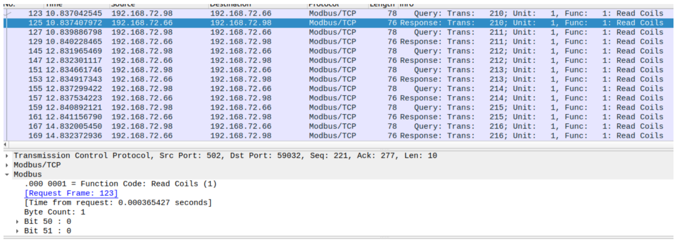
```
root@809ce5e47b9d:/home# nmap -sP 192.168.72.0/24
Starting Nmap 7.93 ( https://nmap.org ) at 2024-06-26 22:07 CEST
Nmap scan report for 192.168.72.66
Host is up (0.000082s latency).
Nmap scan report for 192.168.72.78
Host is up (0.000026s latency).
Nmap scan report for 192.168.72.98
Host is up (0.00089s latency).
MAC Address: 08:00:27:B6:D0:66 (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.72.110
Host is up (0.00038s latency).
MAC Address: 08:00:27:B6:D0:6A (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.72.97
Host is up.
Nmap done: 256 IP addresses (5 hosts up) scanned in 52.50 seconds
root@809ce5e47b9d:/home#
```

Fig. 6: Ping Scan Output

The output (Figure 6) reveals the IP associated with the Master PLC (192.168.72.66), the SCADA Workstation (192.168.72.98), and the two interfaces of the Control Gateway (192.168.72.78/110). Following the reconnaissance, a MitM attack is initiated using ARP poisoning. In this phase, the attacker manipulates the ARP tables of the SCADA Workstation and the Master PLC by sending gratuitous ARP messages. ARP spoofing is performed using the following command:

```
arpspoof -i eth1 192.168.72.110 -rt
↪  192.168.72.98
```

In this command, the *-i* flag specifies the network interface where ARP poisoning activities are conducted, and the *-r* flag enables traffic redirection in both directions (full duplex). With the MitM attack in place, the next phase involves packet sniffing. The objective of this phase is to gather information about the industrial protocol used for exchanging control messages. To achieve this, Wireshark is used to enable the attacker to intercept and display the unencrypted traffic between the SCADA and the General PLC (Figure 7).



```
No.    Time          Source          Destination     Protocol    Length Info
   123 10.837042545  192.168.72.98   192.168.72.66   Modbus/TCP    78    Query: Trans:   210; Unit:   1, Func:   1: Read Coils
   125 10.837407972  192.168.72.66   192.168.72.98   Modbus/TCP    76 Response: Trans:   210; Unit:   1, Func:   1: Read Coils
   127 10.839886798  192.168.72.98   192.168.72.66   Modbus/TCP    78    Query: Trans:   211; Unit:   1, Func:   1: Read Coils
   129 10.840228465  192.168.72.66   192.168.72.98   Modbus/TCP    76 Response: Trans:   211; Unit:   1, Func:   1: Read Coils
   145 12.831965469  192.168.72.98   192.168.72.66   Modbus/TCP    78    Query: Trans:   212; Unit:   1, Func:   1: Read Coils
   147 12.832301117  192.168.72.66   192.168.72.98   Modbus/TCP    76 Response: Trans:   212; Unit:   1, Func:   1: Read Coils
   151 12.834661746  192.168.72.98   192.168.72.66   Modbus/TCP    78    Query: Trans:   213; Unit:   1, Func:   1: Read Coils
   153 12.834917343  192.168.72.66   192.168.72.98   Modbus/TCP    76 Response: Trans:   213; Unit:   1, Func:   1: Read Coils
   155 12.837299422  192.168.72.98   192.168.72.66   Modbus/TCP    78    Query: Trans:   214; Unit:   1, Func:   1: Read Coils
   157 12.837534223  192.168.72.66   192.168.72.98   Modbus/TCP    76 Response: Trans:   214; Unit:   1, Func:   1: Read Coils
   159 12.840892121  192.168.72.98   192.168.72.66   Modbus/TCP    78    Query: Trans:   215; Unit:   1, Func:   1: Read Coils
   161 12.841156790  192.168.72.66   192.168.72.98   Modbus/TCP    76 Response: Trans:   215; Unit:   1, Func:   1: Read Coils
   167 14.832005450  192.168.72.98   192.168.72.66   Modbus/TCP    78    Query: Trans:   216; Unit:   1, Func:   1: Read Coils
   169 14.832372936  192.168.72.66   192.168.72.98   Modbus/TCP    76 Response: Trans:   216; Unit:   1, Func:   1: Read Coils

▸ Transmission Control Protocol, Src Port: 502, Dst Port: 59032, Seq: 221, Ack: 277, Len: 10
▸ Modbus/TCP
▾ Modbus
    .000 0001 = Function Code: Read Coils (1)
    [Request Frame: 123]
    [Time from request: 0.000365427 seconds]
    Byte Count: 1
  ▸ Bit 50 : 0
  ▸ Bit 51 : 0
```

Fig. 7: Wireshark Output

The final phase of the attack involves data tampering on the packets exchanged between the SCADA Workstation and the Master PLC. As depicted in Figure 8, the adversary intercepts a Modbus *Write Single Coil* request from the SCADA HMI, alters the coil value from 1 to 0, and subsequently modifies the Modbus *Read Coil* response from the Master PLC to reflect this unauthorized change.

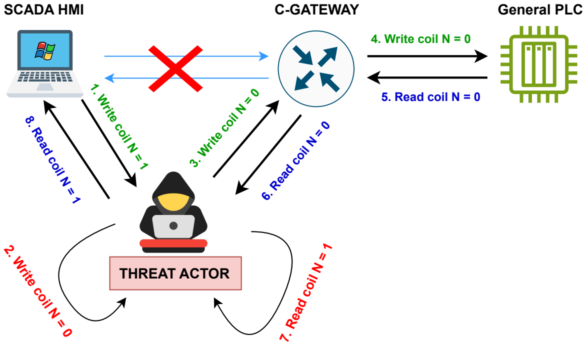


Fig. 8: Man-in-the-Middle Attack Scheme

B. MTD Redundancy Strategy

This subsection delves into the implementation of defense policies employed in order to block the attack described in Subsection VI-A and proactively redirect the malicious traffic in a controlled environment. Specifically, this strategy is fully managed by the SDN controller and operates in two stages.

In the first stage, the SDN controller employs static techniques to determine the validity of a packet. If a packet is found to be invalid, the source host of these packets has likely been corrupted and must be isolated. This validation process relies on Source Address Validation (SAV) and Dynamic ARP Inspection (DAI) to detect anomalies. SAV verifies the authenticity of the source IP address for each packet to ensure it aligns with the expected network configuration. Similarly, DAI validates Address Resolution Protocol (ARP) packets within a network. In detail, DAI enables network administrators to intercept, log, and discard ARP packets with invalid MAC-to-IP address bindings. This combination is effective in identifying ARP poisoning attacks, such as the one described in Subsection VI-A. In our platform, these methods are applied by leveraging the widespread adoption of static network configurations in industrial networks. Due to this static configuration, the SDN controller has prior knowledge of critical network details, such as the IP addresses assigned to hosts and the corresponding switch ports to which they are connected, and can manage Open vSwitches accordingly.

In the second stage, the SDN controller activates an MTD Redundancy strategy to redirect the identified malicious traffic. MTD Redundancy involves deploying multiple copies of a system or network component and allows the seamless replacement of the original one when it comes under attack. Specifically, this is achieved by deploying a copy of the virtualized testbed (with different VLAN IDs) on a separate machine that is then connected to the physical switch through a trunk channel (Figure 9). Once an invalid packet is detected, the SDN controller dynamically modifies the VLAN ID of the corresponding Open vSwitches to isolate the malicious traffic and redirect it to a controlled environment for further analysis.

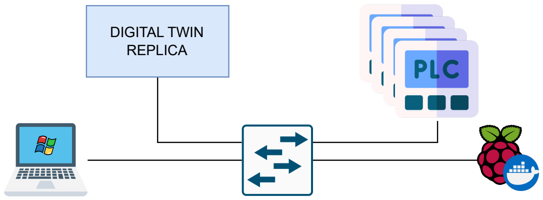


Fig. 9: Digital Twin with its replica

The MTD Redundancy strategy described in this subsection demonstrates the effectiveness of SDN-based defense mechanisms in mitigating ARP poisoning attacks within an industrial network environment. Indeed, the SDN controller is able to detect anomalies and isolate compromised hosts by leveraging SAV and DAI. Additionally, the dynamic configuration of VLAN IDs on access ports ensures that malicious traffic is redirected to a controlled environment without disrupting normal network operations. Lastly, security operators can leverage the Digital Twin Replica to isolate attackers and study their behavior without risking the compromise of critical assets.

VII. Conclusion

This paper introduces a digital twin replicating the EPIC Microgrid testbed aimed at validating cutting-edge SDN-based cyber-defense strategies without disrupting daily industrial operations. At the core of the digital twin are key industrial components, including Programmable Logic Controllers (PLCs), Intelligent Electronic Devices (IEDs), and Supervisory Control and Data Acquisition (SCADA) systems. These components are deployed using a combination of physical hardware, virtual machines, and containers to ensure an accurate simulation of industrial control environments. The interconnected virtual network is constructed entirely using Open vSwitches (OvSs), providing a flexible infrastructure that can be tailored to meet user preferences for simulation fidelity. Additionally, the testbed includes a tool suite designed to collect and analyze relevant data from raw network traffic, containers, and virtual machines. As future work, we plan to integrate the simulation of the physical process to further enhance the fidelity of the digital twin. Additionally, we intend to release a dataset based on the data collected through the testbed.

Acknowledgement

This work has been funded by the project Azione IV.4 - Dottorati e contratti di ricerca su tematiche dell'innovazione - DM 1061 del 10 agosto 2021 (CUP: E65F21003690003).

References

[1] Karen Scarfone, Wayne Jansen, Miles Tracy, "Nist sp 800-123 guide to general server security," accessed: 2024-8-18. [Online]. Available: https://doi.org/10.6028/NIST.SP.800-123

[2] A. J. G. de Azambuja, T. Giese, K. Schützer, R. Anderl, B. Schleich, and V. R. Almeida, "Digital twins in industry 4.0 - opportunities and challenges related to cyber security," *Procedia CIRP*, 2024.

[3] S. Adepu, N. K. Kandasamy, and A. Mathur, "Epic: An electric power testbed for research and training in cyber physical systems security," in *Computer Security: ESORICS 2018 International Workshops, CyberICPS 2018 and SECPRE 2018, Barcelona, Spain, September 6–7, 2018, Revised Selected Papers 2*. Springer, 2019, pp. 37–52.

[4] openvswitch, "openvswitch," accessed: 2024-8-18. [Online]. Available: https://www.openvswitch.org/

[5] NS-UniNa, "Scass-v2," accessed: 2024-8-18. [Online]. Available: https://github.com/NS-unina/SCASS-v2

[6] M. Conti, D. Donadel, and F. Turrin, "A survey on industrial control system testbeds and datasets for security research," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 4, pp. 2248–2294, 2021.

[7] H.-K. Shin, W. Lee, J.-H. Yun, and H. Kim, "Implementation of programmable {CPS} testbed for anomaly detection," in *12th USENIX Workshop on Cyber Security Experimentation and Test (CSET 19)*, 2019.

[8] S. Choi, J. Choi, J.-H. Yun, B.-G. Min, and H. Kim, "Expansion of ICS testbed for security validation based on MITRE ATT&CK techniques," in *13th USENIX Workshop on Cyber Security Experimentation and Test (CSET 20)*. USENIX Association, Aug. 2020.

[9] N. K. Kandasamy, S. Venugopalan, T. K. Wong, and N. J. Leu, "An electric power digital twin for cyber security testing, research and education," *Computers and Electrical Engineering*, 2022.

[10] D. O. B. C. Ekisa and Y. Kavanagh, "Vicsort - a virtualised ics open-source research testbed," *Cyber Research Conference - Ireland*, 2022.

[11] D. Formby, M. Rad, and R. Beyah, "Lowering the barriers to industrial control system security with {GRFICS}," in *2018 USENIX Workshop on Advances in Security Education (ASE 18)*, 2018.

[12] G. Bernieri, F. Del Moro, L. Faramondi, and F. Pascucci, "A testbed for integrated fault diagnosis and cyber security investigation," in *2016 International Conference on Control, Decision and Information Technologies (CoDIT)*, 2016, pp. 454–459.

[13] F. Sauer, M. Niedermaier, S. Kießling, and D. Merli, "Licster–a low-cost ics security testbed for education and research," *arXiv preprint arXiv:1910.00303*, 2019.

[14] P. Čeleda, J. Vykopal, V. Švábenskỳ, and K. Slavíček, "Kypo4industry: A testbed for teaching cybersecurity of industrial control systems," in *Proceedings of the 51st acm technical symposium on computer science education*, 2020, pp. 1026–1032.

[15] G. B. Gaggero, A. Armellin, G. Portomauro, and M. Marchese, "Industrial control system-anomaly detection dataset (ics-add) for cyber-physical security monitoring in smart industry environments," *IEEE Access*, vol. 12, pp. 64 140–64 149, 2024.

[16] NS-UniNa, "Cyberrange-collector," accessed: 2024-8-18. [Online]. Available: https://github.com/NS-unina/CyberRange-Collector