applied sciences

MDPI

Article

A Hybrid Digital-Twin-Based Testbed for Real-Time Manipulation of PROFINET I/O: A Practical Man-in-the-Middle Attack Implementation

Juan V. Martín-Fraile [1,*], Jesús E. Sierra García [1], Nuño Basurto [2] and Álvaro Herrero [2]

[1] Grupo de Investigación en Automatización, Robótica, Control y Optimización (ARCO), Departamento de Digitalización, Escuela Politécnica Superior, Universidad de Burgos, Av. Cantabria s/n, 09006 Burgos, Spain; jesierra@ubu.es

[2] Grupo de Inteligencia Computacional Aplicada (GICAP), Departamento de Digitalización, Escuela Politécnica Superior, Universidad de Burgos, Av. Cantabria s/n, 09006 Burgos, Spain; nbasurto@ubu.es (N.B.); ahcosio@ubu.es (Á.H.)

* Correspondence: jvmartin@ubu.es

Abstract

This study presents a practical methodology for executing Man-in-the-Middle (MitM) attacks on industrial control systems that utilize PROFINET I/O—a communication layer that remains largely underexplored in ICS cybersecurity research. A hybrid digital-twin-based testbed is developed by integrating Siemens S7-1500 and S7-1200 PLCs with a process replica implemented in PCSimu, together with a malicious application that modifies specific process data before it is delivered through the PROFINET I/O channel, enabling controlled falsification of process information in real time. The attacker operates through a Modbus TCP control channel while injecting the manipulated values into the 40-byte Real-Time Class 1 (RTC1) cyclic process-data payload while preserving frame integrity and protocol-level validity indicators. Experimental results show that SDU-level modifications on the 2-ms RTC1 cycle produced deterministic and fully reproducible effects on PLC-level behavior, including forced actuator confirmations and falsified process states, demonstrating the feasibility of both DI- and DO-level manipulation scenarios. Network captures and MSSQL-based event logs provide bit-level correlation between the injected SDU modifications and their impact on the automation sequence, confirming the reliability of the proposed manipulation mechanism. The testbed also supports the systematic generation of labeled datasets for training and evaluating machine-learning-based intrusion and anomaly-detection methods, and offers direct applicability to research, education, and operator-training activities in industrial cybersecurity. Overall, the proposed platform offers a secure, reproducible, and practically applicable environment for vulnerability assessment, attack simulation, and the development of detection techniques in industrial PROFINET networks.

Keywords: PROFINET I/O; RTC1 cyclic real-time communication; Man-in-the-Middle (MitM) attack; SDU manipulation; digital twin testbed; PLC security; Industrial cybersecurity; dataset generation

Check for updates

Academic Editor: Dimitri Konstantas

Received: 4 March 2026
Revised: 27 March 2026
Accepted: 1 April 2026
Published: 3 April 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Industrial Control Systems (ICS) constitute the operational backbone of essential sectors such as manufacturing, energy, and transportation, as well as critical infrastructures

including electric grids, gas pipelines, and water distribution systems. These systems enable the automation of physical processes through the interaction of controllers, sensors, and actuators. The progressive digitalization driven by Industry 4.0 has intensified the convergence between Information Technologies (IT) and Operational Technologies (OT), giving rise to Industrial Cyber–Physical Systems (ICPS) characterized by the tight integration of computational and physical components in real time [1,2].

While these transformations have improved efficiency, monitoring, and remote operation, they have also expanded the attack surface, increasing the exposure of industrial infrastructures to cyber threats. Several studies have shown that traditional SCADA environments exhibit structural vulnerabilities that can be exploited [3–5]. Reference [3] reports significant weaknesses in a SCADA testbed, demonstrating that replay, denial-of-service, and traffic-manipulation attacks can easily compromise Modbus-based systems. Further work [4] confirms that these vulnerabilities persist due to the lack of native authentication, integrity, and encryption mechanisms. More recent analyses [5] show that these deficiencies are still present even in critical infrastructures such as electrical substations, where attackers may capture unencrypted credentials, intercept SCADA traffic, or inject malicious commands via Man-in-the-Middle (MitM) attacks.

Among these threats, MitM attacks are particularly critical because they enable the interception and manipulation of data without clear indicators of intrusion. The absence of built-in security features in commonly used industrial protocols facilitates the injection of false commands or the covert alteration of process signals. Certain MitM attacks can alter operational data without generating anomalies detectable solely through network-traffic analysis, underscoring the need to incorporate physical-process information into detection mechanisms [6,7].

Progress in analysis and detection techniques is further constrained by the limited availability of realistic datasets for ICS environments. Although some works [8] provide large datasets captured from physical testbeds, they primarily focus on DoS, ARP spoofing, or MitM attacks on generic protocols and do not include real-time industrial communication such as PROFINET I/O or the direct manipulation of process signals. This limitation hinders the evaluation and comparison of detection methods under realistic OT conditions.

Moreover, the industrial sector is currently facing concrete protection challenges related to PROFINET-based communication. Industrial Ethernet protocols—including PROFINET—have historically lacked integrated security mechanisms, making them susceptible to data manipulation, spoofing, and Man-in-the-Middle attacks at the protocol level [9]. In parallel, manufacturers, integrators, and operators must now comply with increasingly stringent OT-security regulations, including NIS2, the CER Directive, the Cyber Resilience Act, and the IEC 62443 series, which explicitly mandate authenticity, integrity, and secure configuration of industrial communication systems [10]. These regulatory and technical pressures have motivated the ongoing development of PROFINET Security Classes 2 and 3, which introduce certificate-based mutual authentication, EAP-TLS key establishment, and cryptographic protection of cyclic and acyclic traffic. However, recent studies reveal that secure PROFINET Application Relations (ARs) introduce non-negligible timing overheads—ranging from tens to hundreds of milliseconds depending on the cryptographic configuration—posing challenges for time-critical industrial processes and highlighting the need for empirical evaluation under realistic conditions [11]. Additionally, attempts to extend PROFINET communication beyond the factory network, such as through VPN or cloud-based controllers, expose further limitations and security risks due to PROFINET's reliance on OSI Layer-2 frames and its lack of native end-to-end protection mechanisms [12]. Together, these industrial pain points underscore the relevance of testbeds capable of reproducing PROFINET communication, supporting controlled

signal manipulation, and enabling the assessment of security-related behaviors under operationally realistic scenarios.

Direct experimentation with MitM attacks in live ICS remains limited due to operational risks, high costs, and constraints on process interruption. This has motivated the development of cyber–physical testbeds. However, many existing platforms focus exclusively on network traffic or reproduce simplified scenarios, lacking the capability to actively manipulate input and output signals. Recent contributions highlight the need for platforms that integrate industrial communications with the physical behavior of the process [13–15].

The combination of real industrial hardware and digital twins offers a balanced and safe approach for studying cyber threats. Prior analyses demonstrate the value of such environments for analyzing attacks and detection mechanisms without risking production systems [16,17]. Experimental studies further show that MitM attacks remain feasible even in hardened environments, allowing attackers to modify commands and process data without detection [18].

Within this context, the present work introduces a hybrid testbed based on a digital twin capable of executing practical Man-in-the-Middle (MitM) manipulation techniques in industrial environments. The platform integrates a physical PLC (PLC 1), a second PLC configured as a PROFINET I/O iDevice (PLC 2), and a dedicated manipulation module that injects falsified values into the RTC1 process-data payload. Rather than operating as a physical network intermediary, the attack module cooperates with PLC 2 to alter the process truth before it is delivered to PLC 1, enabling real-time modification of sensor readings or actuator states without placing the physical infrastructure at risk.

Unlike contributions focused solely on detection techniques, this work centers on an experimental infrastructure that enables direct manipulation of process behavior through selective modification of I/O signals, with full control over the duration, persistence, and scope of the injected values. This capability makes the platform a versatile tool for applied research, dataset generation, and industrial cybersecurity training.

The main technical contributions of this work are as follows:

(CT1) A hybrid industrial testbed combining real hardware (Siemens PLCs (Siemens AG, Munich, Germany) and a PROFINET iDevice) with a digital twin, enabling safe and realistic experimentation on PROFINET I/O communications.

(CT2) An active MitM attack module capable of manipulating industrial signals in real time through direct modification of the RTC1 Service Data Unit (SDU)—the PROFINET I/O payload containing process-data bytes together with the standard IOPS/IOCS status indicators—without altering network traffic.

(CT3) A set of representative attack scenarios that demonstrate the scope, flexibility, and practical relevance of the proposed platform.

(CT4) Integrated monitoring through port mirroring, enabling the capture and analysis of PROFINET I/O traffic during attacks using tools such as Wireshark, and facilitating the study of how signal manipulation affects PROFINET cycle timing—a topic rarely addressed in the current literature.

1.1. Scientific Contribution

This work provides a scientific contribution at three complementary levels. First, it empirically demonstrates that semantic manipulation of PROFINET I/O at the SDU layer can alter the behavior of an industrial process without affecting RTC1 cycle timing, protocol-level validity indicators (IOPS/IOCS), or availability metrics. This reveals a critical blind spot in current ICS integrity validation mechanisms, which ensure structural and temporal correctness but do not verify the semantic coherence of process data values.

Second, the study introduces a generalizable experimental methodology that integrates physical process modeling, real hardware execution, SDU-level payload manipulation, and multi-layer validation (RTT, throughput, cycle timing, and GEMMA-based behavioral analysis). This methodology is not tied to the specific PLC–digital-twin pair used here and can be adapted to other industrial communication protocols (e.g., PROFIBUS, INTERBUS, DeviceNet, etc.), as long as an intermediate control node provides access to process data manipulation.

Third, the proposed platform offers a reproducible research framework capable of generating synchronized physical, logical, and network-level datasets for evaluating anomaly-detection approaches, semantic-validation techniques, and intelligent intrusion-detection mechanisms. Taken together, these capabilities position the testbed as a scientifically grounded instrument that enables the development of intelligent and machine-learning-based protection strategies for industrial cyber-physical systems.

1.2. Research Questions

Building on the gaps identified above, this work addresses the following research questions:

- RQ1. Does the proposed hybrid testbed provide sufficient flexibility to reproduce realistic PROFINET I/O communication and enable controlled Man-in-the-Middle (MitM) manipulation of industrial signals without risking physical infrastructure?
- RQ2. Does the execution of MitM attacks in the proposed testbed introduce measurable deviations in standard communication metrics—such as ICMP round-trip times, throughput, or PROFINET cycle timing—compared to baseline operation?
- RQ3. Is the testbed capable of generating reproducible datasets suitable for supporting anomaly detection, attack analysis, and training activities in industrial cybersecurity?

The remainder of this article is structured as follows: Section 2 reviews related work on ICS cybersecurity and MitM attack modeling. Section 3 describes the architecture of the testbed. Section 4 presents the implementation of the MitM attack, including the attack logic, the preceding reconnaissance phase, and the experimental scenarios implemented in the testbed. Section 5 analyzes the experimental results. Section 6 discusses the findings and their implications. Finally, Section 7 summarizes the conclusions and outlines future research directions.

2. Related Work

The increasing digitalization of Industrial Control Systems (ICS) and SCADA environments has substantially expanded their exposure to cyberattacks, particularly those targeting or intercepting communication between critical devices. For more than a decade, studies have shown that SCADA systems exhibit structural vulnerabilities in both protocols and architectures, mainly due to the absence of authentication, integrity, and encryption mechanisms in widely used protocols such as Modbus TCP. Early analyses demonstrated multiple exploitable weaknesses in SCADA testbeds, showing that denial-of-service, replay, and traffic-manipulation attacks can easily compromise Modbus-based systems [3]. Subsequent work confirmed that these vulnerabilities persist because the protocol lacks essential security primitives [4].

Recent literature has examined these deficiencies more comprehensively. A systematic assessment of Modbus TCP from an ICT perspective highlights that the absence of cryptographic protections enables impersonation, frame manipulation, and tampering with industrial traffic [4]. These findings reinforce the notion that Modbus TCP continues to represent one of the most vulnerable components in modern ICS architectures.

In parallel, the scientific community has progressed toward developing testbeds capable of reproducing realistic attack scenarios, since experimentation on operational

industrial systems carries high risk, high cost, and operational constraints. Recent work demonstrates that such environments are essential for assessing attack impact, generating reliable datasets, and validating detection strategies [13–15]. One line of research presents cyber–physical testbeds that integrate both physical-process behavior and industrial communication in real time, while other contributions combine electrical simulators with Modbus-based communication layers to study operational impacts under representative conditions.

Efforts toward high-fidelity experimental platforms continue to advance the field. A recent physical testbed built on Mitsubishi R04ENCPU PLCs and SCALANCE industrial switches executed seven categories of attacks—including DoS, ARP spoofing, replay, scanning, and ARP-based MitM—to produce a realistic dataset exceeding 2.6 million labeled records [8]. The study showed that virtual datasets fail to capture the temporal, electrical, and communication dynamics of operational ICS, reinforcing the need for physical testbeds when evaluating advanced cybersecurity techniques.

While public datasets such as ICSCASD MPLC [8] provide valuable flow-based network traffic captured from real industrial testbeds—including TCP/UDP communications, ARP-based attacks, and multiple intrusion scenarios—they operate at the packet and flow-statistics level and do not expose real-time industrial I/O dynamics. In contrast, the dataset generated in this work targets a different analytical layer: cyclic PROFINET I/O RTC1 communication, SDU-level payloads, and the direct manipulation of DI/DO signals synchronized with the evolution of the physical process. This cyber–physical alignment, supported by a hybrid digital-twin plus real-PLC environment, provides process-level features not present in existing public datasets and enables the study of fine-grained, real-time manipulation attacks on PROFINET-based industrial I/O.

Among the threats affecting ICS, Man-in-the-Middle (MitM) attacks remain especially critical due to their ability to alter operational data flowing between PLCs, HMIs, and field devices while preserving protocol-level validity indicators, often without generating clear signs of compromise. Existing studies report that MitM and replay attacks pose major challenges for detection methods relying solely on network analysis [6], and later work confirms that MitM attacks can modify process values without causing detectable anomalies at the network level. These findings underscore the need for detection approaches that combine network-traffic information with process-level data [7].

MitM attacks have also been validated experimentally in real industrial environments. Experiments conducted on hardened systems have also shown that Man-in-the-Middle techniques remain feasible even in environments with encrypted control traffic, demonstrating operational impact and system response [18]. This provides strong evidence that MitM attacks remain feasible even in systems equipped with additional protection mechanisms.

Other contributions have examined DoS and basic MitM-related attacks on real PLCs in energy settings. Analyses performed on Siemens S7-1200 and Schneider M241 devices showed that both controllers become inaccessible under port-saturation attacks and that the lack of encryption enables the extraction of sensitive information using tools such as Wireshark [19]. However, these studies focus on low-level network effects and do not address selective manipulation of process data or alterations in cyclic, real-time industrial protocols such as PROFINET I/O.

In distributed energy systems, the impact of MitM attacks has also been examined in DERMS platforms, where manipulation of critical data was shown to cause operational instability in modern infrastructures [5]. Although focused on a different domain, these findings reinforce the relevance of studying integrity-focused attacks in industrial environments.

Beyond attack characterization, several studies propose detection mechanisms. Random-Forest-based approaches have been used to identify MitM attacks in advanced

metering infrastructures [20], while LSTM autoencoders have demonstrated strong performance in detecting anomalies even when manipulated values appear legitimate at the network level [7]. Additional work reports superior performance from tree-based models when analyzing real SCADA datasets [21]. However, these techniques are generally evaluated on protocols such as Modbus/TCP and do not address semantic manipulation of cyclic, real-time industrial traffic such as PROFINET I/O.

Overall, the reviewed literature highlights three main trends:

1. Widely deployed industrial protocols—particularly Modbus/TCP—continue to exhibit significant vulnerabilities to manipulation and lack built-in integrity mechanisms.
2. Hybrid and cyber–physical testbeds have become essential for safe experimentation and dataset generation.
3. Research is increasingly combining network-level information with process-level behavior and machine-learning techniques to enhance detection capabilities.

However, none of these contributions address PROFINET I/O in its real-time RTC1 modality, and no prior work evaluates Man-in-the-Middle attacks that selectively manipulate the SDU before cyclic transmission. This gap motivates the hybrid platform proposed in this study.

To address this limitation, the proposed platform enables semantic manipulation of PROFINET I/O at the SDU level using real hardware and cyclic I/O exchange. Unlike prior approaches, the testbed integrates physical-process modeling, controlled signal manipulation, and synchronized physical, logical, and network-level monitoring, providing a unique foundation for evaluating the impact of semantic perturbations within real-time industrial communication. Furthermore, while existing detection-oriented studies rely mostly on high-level flow statistics or packet traces, the proposed platform generates aligned datasets that capture process semantics, controller behavior, and PROFINET RTC1 traffic simultaneously. These characteristics differentiate the proposed approach from State-of-the-Art systems and contextualize its contribution within current ICS cybersecurity research.

3. Testbed Architecture

The architecture of the testbed follows a hybrid, open, and flexible approach that integrates real industrial hardware, a digital-twin representation of the process implemented in PCSimu [22], and a logical manipulation module that operates on the signals exchanged between the twin and PLC 2. The main objective is to provide a safe and reproducible environment for evaluating Man-in-the-Middle attacks that target the integrity of process data, as well as for dataset generation and training activities. The system is organized into three functional layers—virtualized process, distributed control, and supervision—whose interaction enables the recreation of realistic operating conditions without jeopardizing any physical infrastructure. Figure 1 shows the overall architecture of the testbed.

3.1. System Overview

The testbed is organized into three functional levels that reproduce the interaction between the virtualized process, the distributed control layer, and the system-level supervision.

Level 0 consists of the digital twin and PLC 2, configured as a PROFINET I/O iDevice, which together form the virtual representation of the plant. The twin reproduces the dynamic behavior of the process, and exchanges simulated I/O values with PLC 2 through PUT/GET (S7 communication). Using these signals, PLC 2 generates the process data later delivered to PLC 1 over the PROFINET I/O channel, emulating a real decentralized device. In addition, PLC 2 runs the standard MB_SERVER block, exposing Modbus registers that the Python application can read and write. These registers support the logical MitM mechanism: in each PLC cycle, internal logic merges the real signals from PCSimu with the

manipulated values supplied by Python, selecting which bits are forwarded depending on the attack configuration.

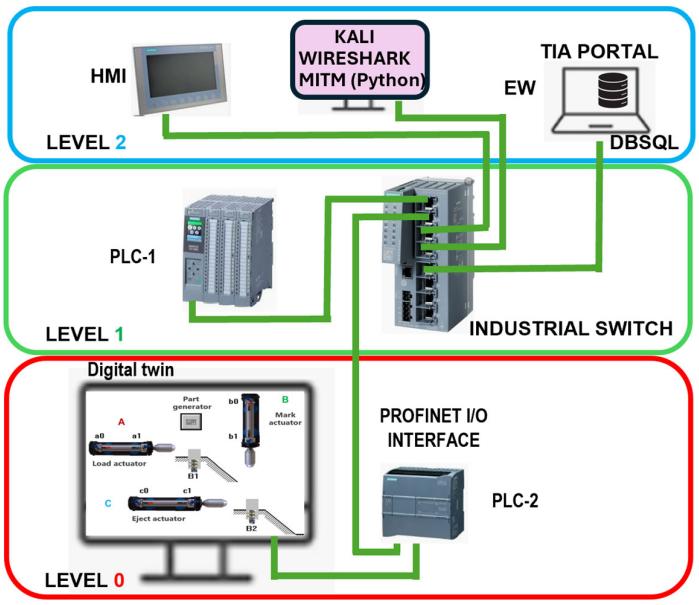


Figure 1. General Testbed Architecture.

Level 1 consists solely of PLC 1, which acts as the main controller of the process. It executes the automation logic using the signals received from PLC 2 over the PROFINET I/O channel, without performing additional integrity checks. As a result, any logical alteration introduced at PLC 2 is directly reflected in PLC 1's control cycle.

Level 2 integrates the supervision and monitoring components: the HMI for visualization, the engineering workstation with TIA Portal [23], the optional MSSQL server for event logging, and analysis tools such as Kali Linux and Wireshark used during the discovery and evaluation phases. This layer also hosts the Python application that implements the logical MitM module, which communicates with PLC 2 through Modbus TCP. By reading and writing the registers exposed by MB_SERVER, the application cooperates with the PLC logic to determine which manipulated or real values are sent to PLC 1—and thus influence its perception of the process. This layer also contains the industrial switch that interconnects all devices in the testbed.

This layered design reproduces a realistic ICS environment by combining industrial hardware, an operational digital twin, and a controlled mechanism for logical signal manipulation. Together, these elements ensure safe experimentation and enable precise analysis of the effects of SDU-level attacks.

Real-Time Performance Parameters

To ensure authenticity and reproducibility, the testbed operates under a set of real-time execution parameters required to characterize its behavior. The PROFINET RTC1 communication between PLC 1 and PLC 2 uses a 2 ms update cycle, corresponding to the default value automatically assigned when the S7-1200 is configured as an I-Device. During operation, PLC 1 (S7-1500) executes the OB1 cyclic program with a typical scan time of 5.4 ms (range: 3.6–5.4 ms, with occasional peaks caused by communication servicing), while PLC 2 (S7-1200) exhibits a typical scan time of 4.0 ms (range: 2–5 ms).

The digital-twin module (PCSIMU) exchanges process values with PLC 2 through PUT/GET communication services, which only requires enabling the option "Permit access with PUT/GET communication from remote partner" in the PLC settings. PCSIMU does not expose a deterministic internal execution cycle. Therefore, the exact timing between

updates cannot be retrieved. In practice, PCSIMU simulates the process logic: when PLC 1 sends commands to the plant (e.g., YA+) by energizing the solenoid valve that controls an actuator, the virtual actuator in PCSIMU performs the corresponding movement according to its configured valve type and speed, and once the end position is reached, the associated digital-input signal (a1) is activated. The time required for each transition is not deterministic and may vary between cycles—just as in real industrial installations, where pneumatic or mechanical dynamics introduce natural fluctuations. What remains relevant is the sequence integrity, while the PROFINET RTC1 channel enforces the deterministic 2 ms timing under which the MitM scenarios are executed.

3.2. Hardware Components

The testbed integrates the following physical elements:

- PLC 1 (main controller): Acts as the central controller and executes the automation logic based on the signals provided by PLC 2 through the PROFINET I/O channel. Since it performs no additional integrity checks, any modification introduced upstream is directly reflected in its control cycle.
- PLC 2 (PROFINET iDevice): Configured as a PROFINET I/O device, it cyclically exchanges simulated sensor and actuator states with the digital twin via PUT/GET (S7 communication), emulating the behavior of a decentralized peripheral. PLC 2 also runs the MB_SERVER block, exposing Modbus TCP registers to the Python application. By reading and writing these registers, the Python module cooperates with the PLC logic to determine which real or manipulated values are inserted into the process-data payload ultimately delivered to PLC 1.
- Industrial switch: Provides the Ethernet communication backbone interconnecting PLC 1, PLC 2, the HMI, the engineering workstation, and the Python module. It is not used as an attack-insertion point, but simply as the physical networking component required for coordinated system operation.
- HMI: Enables process supervision, status visualization, and operator interaction. Its display reflects either the real or manipulated behavior of the system, depending on the values propagated through the controllers during each scenario.

Although these hardware components form the physical foundation of the testbed, their behavior depends on a set of software modules that virtualize the process, coordinate communication between controllers, and implement the logical MitM attack. These modules are described in the following section.

In addition to these industrial devices, the testbed also includes three host PCs (Engineering Workstation, Digital-Twin Host, and Attacker PC), which support the execution of TIA Portal, PCSIMU, event-logging services, and the Python-based MitM module.

To ensure full reproducibility, Table 1 lists the complete hardware configuration of the testbed, including the exact models, manufacturer information, and roles of all physical devices described in Section 3.2. This information provides the necessary level of detail for replicating the hardware setup used in the experiments.

Table 1. Hardware components used in the testbed.

Component	Model	Manufacturer	Role in the Testbed
PLC 1	6ES7512-1CK01-0AB0	Siemens AG, Munich, Germany	Main Controller
PLC 2	6ES7215-1BG40-0XB0	Siemens AG, Munich, Germany	PROFINET I-Device

Table 1. *Cont.*

Component	Model	Manufacturer	Role in the Testbed
HMI Panel	6AV2 123-2JB03-0AX0	Siemens AG, Munich, Germany	HMI panel
Switch	6GK5208-0BA00-2AC2	Siemens AG, Munich, Germany	Industrial Ethernet
Engineering PC	PC (Intel i7, 32 GB RAM)	-	TIA Portal, MSSQL
Digital Twin Host	PC (Intel i5, 8 GB RAM)	-	Runs PCSIMU
Attacker PC	PC (intel i7, 16 GB RAM)	-	Python MitM, Wireshark

3.3. Software Components

The system is complemented by the following logical components, distributed across three independent host machines as depicted in Figure 1:

- Engineering Workstation: Host platform used for TIA Portal programming, network diagnostics, PROFINET monitoring, and execution of tools such as MSSQL Server Express for event-log storage, data analysis, and dataset generation. Although it runs on a standard PC rather than industrial hardware, it provides the supervisory and analysis environment required for experimentation.
- Digital twin: Executed on a dedicated PC using the industrial simulation tool PCSimu, it reproduces the dynamic behavior of the process and updates the process image of PLC 2 through PUT/GET (S7 communication). Through this mechanism, PCSimu provides virtual sensors and actuators whose signals are transferred to PLC 2 as if they originated from a real peripheral device. These signals—either genuine or manipulated—are then forwarded to PLC 1 via PROFINET I/O.
- Man-in-the-Middle (MitM) module–Python application (Attacker PC): Run on a separate host with network access to PLC 2, the Python application implements a logical MitM mechanism without acting as a network proxy between PLC 1 and PLC 2. It communicates with PLC 2 over Modbus TCP through the MB_SERVER block, reading and writing the registers that contain the packaged process signals. In collaboration with internal PLC 2 logic, the module determines whether the values received from PCSimu are propagated unchanged or replaced by manipulated values (e.g., bit masks or forced states). This modification alters the "process truth" that PLC 2 subsequently forwards to PLC 1 over PROFINET I/O.
- Monitoring and Reconnaissance Tools (Attacker PC): Tools such as Wireshark, Nmap/Kali Linux, and diagnostic utilities are executed on the attacker PC. They support reconnaissance, validation of the manipulation mechanism, and packet-level analysis of the PROFINET I/O exchange through mirrored-port captures.

These software components provide the supervisory, simulation, manipulation, and monitoring capabilities required for executing and analyzing the attack scenarios.

Table 2 summarizes the software components and version information required to reproduce the experimental environment, complementing the detailed description provided in Section 3.3. These specifications ensure that the testbed configuration and the MitM experiments can be replicated accurately.

Table 2. Software components used in the testbed.

Component	Version	Platform/OS	Role in the Testbed
TIA Portal	V18 Update 5	Windows 11	PLC engineering
PCSIMU	V3.0	Windows 10	Digital twin
Python	3.13.5	Windows 10	MitM module
Pymodbus	3.11.4	Python	Modbus client
MB_SERVER block	Standard lib.	PLC S7 1200	Modbus server
MSSQL Server Express	2019	Windows 11	Event logging
SSMS	20.2.30.0	Windows 11	SQL administration
Wireshark	4.6.3	Windows 10	Network capture
Kali Linux tools	Rolling	Linux	Reconnaissance

3.4. Interfaces and Protocols

The testbed integrates several industrial communication channels that accurately reproduce the behavior of a real ICS environment. Each channel fulfills a specific role within the data flow between the digital twin, the controllers, and the Python-based manipulation module.

- PROFINET I/O (PLC 1 ↔ PLC 2): The communication between PLC 1 and PLC 2 via PROFINET I/O constitutes the cyclic I/O channel on which the execution of the control logic relies. In this configuration, PLC 1 acts as the IO-Controller and PLC 2 as the iDevice, meaning that PLC 1 receives the signals generated or manipulated at PLC 2 as if they originated from a real decentralized peripheral.
- RTC1 cyclic communication and the Provider–Consumer model: The PROFINET I/O exchange is based on 40-byte RTC1 telegrams whose fields follow the Provider–Consumer model. Each submodule transmits process data together with two status indicators: the IO-Provider State (IOPS) and the IO-Consumer State (IOCS). IOPS indicates the validity of the data provided by the sender, whereas IOCS specifies whether the receiver has processed it correctly. As a direct consequence of this model, the SDU layout depends on the direction of the flow: PLC 2 → PLC 1: IOPS–Process Data–IOCS. PLC 1 → PLC 2: Process Data–IOPS–IOCS. This organization matches the RTC1 captures obtained in the testbed (Figure 2) and corresponds to the cyclic-update mechanism described in the official PROFINET IO-Device stack specification [24]. Since IOPS/IOCS are automatically managed by the controllers, the Python module can modify only the process-data bytes without affecting protocol-level validity.
- Modbus TCP (Python module ↔ PLC 2): The Python application communicates with PLC 2 through the MB_SERVER block, which exposes a set of Holding Registers containing the packaged process signals. By reading and writing these registers, the application injects manipulated values before PLC 2 assembles the RTC1 payload and forwards it to PLC 1, forming the insertion point of the logical MitM mechanism.
- S7 PUT/GET Communication (PCSimu ↔ PLC 2): PCSimu exchanges virtualized sensor and actuator states with PLC 2 using PUT/GET operations (S7 Comm), reading and updating the PLC's process image through standard I/O addresses (e.g., I0.0, Q0.0). This mechanism allows the digital twin to emulate process behavior without requiring additional physical equipment.

The digital twin is not synchronized with a physical plant but instead operates in a simulated environment, allowing controlled evaluation of SDU-level manipulations and anomalous behaviors in a safe and reproducible setting.

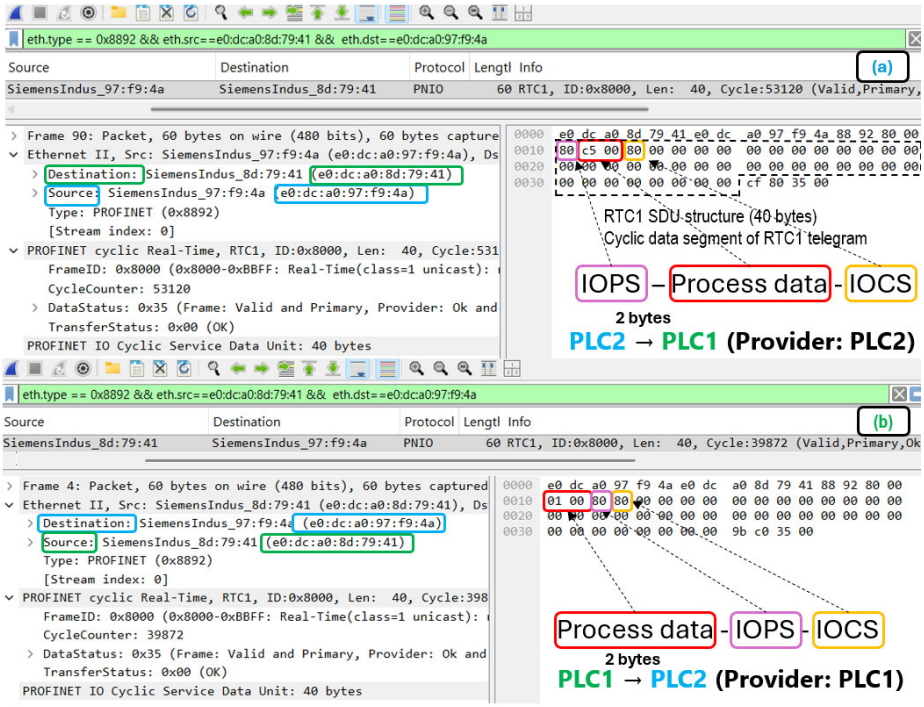


Figure 2. RTC1 cyclic telegrams in both directions: PLC2→PLC1 (**a**) and PLC1→PLC2 (**b**).

3.5. Communication Flow and MitM Insertion Point

The operation of the testbed during a MitM experiment is structured around a communication flow in which the manipulation does not occur on the network between the controllers, but rather at the point where PLC 2 merges the signals originating from the digital twin with the values supplied by the Python application. This results in a logical MitM, where the alteration is applied to the "process truth" before it is transmitted to PLC 1 over PROFINET I/O.

The operational sequence proceeds as follows. First, PCSimu updates the process-image values in PLC 2 through PUT/GET (S7 communication), writing virtualized sensor and actuator states directly into its memory. Next, the Python application accesses the Holding Registers exposed via MB_SERVER to read and modify the data words where the process signals are packaged. PLC 2 then executes an internal logic block that combines, cycle by cycle, the values received from PCSimu with the manipulated values coming from Python. Using bit masks, the PLC determines which bits are preserved and which are replaced.

The resulting signals—whether genuine or altered—are then sent to PLC 1 via PROFINET I/O, where they are interpreted as if they originated from a real decentralized peripheral. Since PLC 1 does not apply additional integrity checks, it incorporates these values directly into its control logic with no ability to detect prior manipulation.

This design, based on coordinated cooperation between the Python module and PLC 2, enables a MitM attack without modifying the network topology or intercepting traffic between PLCs. By acting directly on the process data exchanged between the digital twin and PLC 2, the system reproduces the effect of an integrity-manipulation attack in a precise and controlled manner, while preserving the determinism and stability of the experimental environment.

Figure 3 graphically summarizes this flow, contrasting normal operation with the scenario in which the logical MitM mechanism is active.

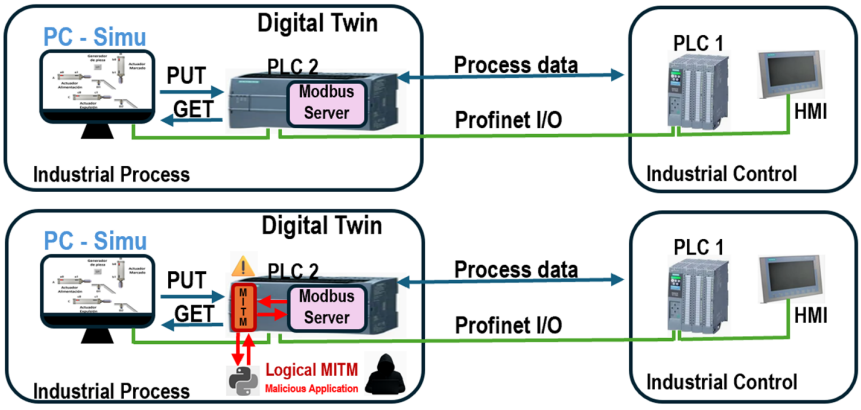


Figure 3. Communication flow between PCSimu, PLC 2, and PLC 1 during normal operation (**top**) and under a logical MitM attack (**bottom**).

3.6. Generalizable Framework for Semantic Manipulation and Multi-Layer Validation

Building on the architectural elements described above, the proposed experimental setup can be abstracted into a generalizable framework that extends beyond the specific PLC–digital-twin configuration used in the testbed. This framework is organized into four methodological pillars that can be replicated across a wide range of industrial control environments.

- Physical-Process Modeling or Acquisition: The framework begins with a representation of the physical process, which may be implemented using a digital twin, a hardware-in-the-loop simulator, or a real industrial asset. Although the present implementation focuses on digital input/output signals, the methodology is not restricted to this subset; it can be extended to any process variable, whether digital, analog, or structured, whose semantics influence the control sequence.
- Intermediate Control Node for Data Manipulation: A key element of the framework is the presence of an intermediate controller capable of merging genuine process signals with externally supplied manipulated values. In the proposed testbed, this role is fulfilled by PLC 2, but equivalent functionality can be achieved with any device that offers cyclic I/O exchange and provides programmatic or register-level access to its process image. This abstraction allows the methodology to be adapted to other industrial communication protocols (e.g., PROFIBUS, INTERBUS, DeviceNet), as long as the intermediate node can intercept and selectively rewrite I/O data before it reaches the primary controller. As the framework is not tied to a particular vendor or control architecture, the primary controller may be implemented using PLCs or PACs from manufacturers such as Allen-Bradley, Siemens, Omron, and others.
- Semantic Manipulation Layer: The logical MitM mechanism operates at the semantic level, modifying I/O data without altering the underlying communication frames. This layer can be implemented through Modbus/TCP, vendor-specific APIs, or any interface that enables controlled injection of process-data values. The abstraction relies on the ability to manipulate the process "truth" upstream of the consumer controller, independent of the communication stack used for cyclic exchange.
- Multi-Layer Monitoring and Validation: To ensure reproducibility and scientific rigor, the framework integrates physical-state logging, controller-behavior monitoring, and network-traffic acquisition. This multi-layer monitoring and validation approach provides synchronized evidence of both the semantic manipulation and its consequences on the industrial process. In addition, structured event logging through a database system (e.g., MSSQL, MySQL, PostgreSQL, or an equivalent solution) enables precise correlation between GEMMA transitions, manipulated values, and PROFINET RTC1 traffic. The design is independent of the specific controller or industrial communi-

cation protocol employed, enabling the framework to support comparative studies across heterogeneous ICS environments.

Overall, this generalizable framework formalizes the methodology introduced in this work and demonstrates how the proposed architecture can be transferred to different control platforms, communication buses, and cyber-physical system configurations, both industrial and non-industrial. In this way, it provides a reusable foundation for the systematic study of semantic manipulation, anomaly detection, and intelligent protection mechanisms across a wide range of environments.

4. Man-in-the-Middle Attack Implementation

This section describes how the Man-in-the-Middle (MitM) attack is implemented within the hybrid digital-twin-based testbed developed for the industrial part-marking system. The attack does not intervene in the network between PLC 1 and PLC 2; instead, it operates as a logical MitM applied at the point where PLC 2 combines the virtualized signals from the digital twin with the manipulated values provided by the Python application. In this way, the alteration affects the "process truth" received by PLC 1 over PROFINET I/O while leaving the underlying communication layers intact.

In a real system, an attacker would achieve this level of access only after an initial compromise of the OT network or an engineering workstation. Once inside the subnet, PLC 2 becomes discoverable through standard reconnaissance techniques such as ARP scanning, TCP SYN probing, or PROFINET DCP identification. If a maintenance service such as Modbus TCP (port 502) remains unintentionally exposed and configured to allow write operations, the attacker can legitimately interact with the PLC's memory areas. This represents a realistic post-compromise scenario frequently observed in industrial environments and provides the required level of access for the logical MitM manipulation demonstrated in our testbed.

Building on the architecture detailed in Section 3, the description below covers the operation of the process, the digital signals involved, the target of the manipulation, and the mechanism through which the attack logic selectively modifies input and output values. The activation logic, signal-mask configuration, and coordination between the Python module and PLC 2 are also detailed to illustrate how controlled falsification of operational data is achieved.

4.1. Industrial Process Description

The process corresponds to a part-marking system represented through a digital twin developed in PCSimu. The digital twin provides a functional simulation of the process, generating virtual sensor and actuator signals with timing and sequencing behavior comparable to an industrial cell, but without reproducing the full dynamics of a real physical system.

As illustrated in Figure 4, the process includes three actuators (A, B, and C), each equipped with magnetic position sensors (a0, a1, b0, b1, c0, c1), together with two inductive detectors (B1 at the loading station and B2 at the marking station) and a workpiece generator that feeds the production cycle. The digital twin exchanges virtualized sensor and actuator states with PLC 2 via PUT/GET (S7 communication), with PLC 2 configured as a PROFINET I/O iDevice that forwards these values to PLC 1, which acts as the main controller. This configuration reproduces a decentralized control architecture suitable for controlled experimentation in a safe environment.

Interaction with the system is performed through an HMI that allows adjustment of operational parameters—such as marking time, number of impacts per workpiece, and batch size—selection between manual and automatic modes, issuing extend/retract

commands to the actuators (OUT_x/IN_x, with x = A, B, C), and controlling start, stop, reset, rearm, and acknowledge operations. The system also incorporates safety elements, including an emergency stop and a four-color signal beacon. From a control-system perspective, the process follows the GEMMA methodology, with stop (A), operation (F), and fault (D) modes.

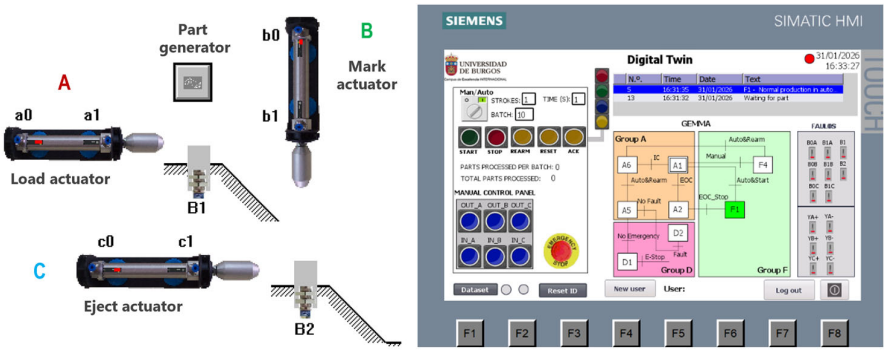


Figure 4. PCSimu digital twin (**left**) and corresponding HMI panel (**right**).

To facilitate a better understanding of the process, Figure 5 presents the partial GRAFCET corresponding to the automatic normal production operating mode, in accordance with the GEMMA guide.

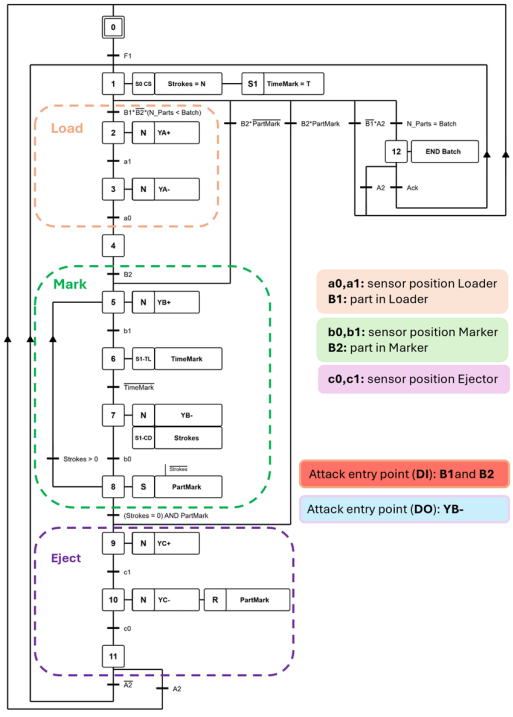


Figure 5. Partial GRAFCET representation of the automatic production cycle (F1).

In this GRAFCET, it can be observed that the initial step remains waiting for mode F1. Once mode F1 is activated, step 1 proceeds to load the parameters set by the operator for the part (strokes and time mark).

Next, the system evaluates the state of the marking unit to determine whether it is necessary to load a part into the marking station (detector B2 inactive) and, in that case, when a part is available at the loading station (detector B1 active), it issues the YA+ command to the solenoid valve so that actuator A advances and performs the part loading.

Once confirmation of the forward position is received (sensor a1 active), the controller withdraws YA+ and commands the return to the initial position via YA−.

Subsequently, when the part is detected by B2 at the marking station, the marking phase begins by activating YB+ until its magnetic sensor b1 is triggered. At that moment, YB+ is withdrawn, and the actuator remains in that position until the configured marking time elapses; afterwards, YB− is commanded to return to the initial position (b0 active), either to execute a new impact or to proceed to the part evacuation phase by means of YC+ and YC−.

Once the part has been evacuated, the controller evaluates whether the cycle should continue until the batch is completed or whether there is a stop request at the end of the cycle from the operator (GEMMA state A2).

As can be observed, based on the program logic and the information received from sensors and detectors (inputs), the controller issues commands (outputs) to the solenoid valves in order to actuate the actuators. These I/O signals constitute the target of the MitM attack described in the following sections, whereas HMI-level commands remain outside the scope of manipulation. Nevertheless, Figure 5 highlights the input and output signals that are targeted by the attack.

4.2. Attack Motivation and Target I/O Signals

The logical MitM attack is oriented toward the selective manipulation of the digital input (DI) and digital output (DO) signals exposed by PLC 2 through its Modbus TCP server (MB_SERVER). The DI values correspond to virtual sensor states generated by the digital twin, while the DO values originate from PLC 1 as actuator-command signals. Both sets of signals are packaged in PLC 2 and can be modified by the Python application before PLC 2 generates the final values to be transmitted through the PROFINET I/O channel—either incorporating DI values received from the digital twin or forwarding DO commands originating from PLC 1, depending on the direction of the flow. The specific DI and DO signals that can be targeted by the attack are summarized in Table 3.

Table 3. Digital signals of the process vulnerable to MitM manipulation.

Process Data	Symbol	Direction	Type	Comment
Digital Input (DI)	a0	I 0.0	Bool	Loader backward (actuator A)
Digital Input (DI)	a1	I 0.1	Bool	Loader forward
Digital Input (DI)	b0	I 0.2	Bool	Marker up (actuator B)
Digital Input (DI)	b1	I 0.3	Bool	Marker down
Digital Input (DI)	c0	I 0.4	Bool	Ejector backward (actuator C)
Digital Input (DI)	c1	I 0.5	Bool	Ejector forward
Digital Input (DI)	B1	I 0.6	Bool	Part detector at the loader
Digital Input (DI)	B2	I 0.7	Bool	Part detector at the marker
Digital Output (DO)	YA+	Q 0.0	Bool	Loader forward
Digital Output (DO)	YA−	Q 0.1	Bool	Loader backward
Digital Output (DO)	YB+	Q 0.2	Bool	Marker down
Digital Output (DO)	YB−	Q 0.3	Bool	Marker up
Digital Output (DO)	YC+	Q 0.4	Bool	Ejector forward
Digital Output (DO)	YC−	Q 0.5	Bool	Ejector backward

The digital inputs from magnetic position sensors (a0, a1, b0, b1, c0, and c1; see Figure 4), together with inductive detectors B1 and B2, allow PLC 1 to determine in real time the position of the actuators and the presence or absence of workpieces in the loading and marking zones. The digital outputs (Yx+ and Yx−, with x = A, B, and C), associated with the extension and retraction commands of the actuators, directly determine the evolution of the automated sequence and contribute to the operational safety of the system.

The manipulation of these signals is particularly critical for three main reasons:

1. Immediate impact on the control logic: Altering a single bit can cause PLC 1 to interpret a nonexistent state, leading to unintended movements, incorrect transitions between stages, or even interrupting the work cycle.
2. Low local detection capability: In a decentralized architecture, PLC 1 fully trusts the data provided by PLC 2. Since Modbus/TCP lacks native mechanisms for authentication, integrity verification, or cross-checking, a logical MitM can remain unnoticed as long as the communication format remains valid and the cyclic exchange remains stable.
3. High potential for operational damage: Manipulating sensors or actuators can cause anything from automatic-sequence blockages to mechanical collisions or damage to the workpiece or the machine itself, depending on the experimental scenario.

Taken together, these digital I/O signals represent a high-value target for an attacker aiming to compromise the integrity, coherence, and reliability of the automated process. Their controlled manipulation within the testbed enables the safe evaluation of the effects of a MitM attack in a decentralized industrial environment.

4.3. MitM Attack Logic

The attack is not physically inserted into the network between PLC 1 and PLC 2; instead, it is implemented as a logical MitM on the PCSimu ↔ PLC 2 signal path. This mechanism is achieved through the interaction of two coordinated components:

- A Python application that reads and writes values via Modbus/TCP to the MB_SERVER block in PLC 2.
- An internal logic block in PLC 2 that determines, cycle by cycle, whether the signals exposed to the PROFINET I/O exchange with PLC 1 correspond to the values received from the digital twin or to the values manipulated by Python.

The manipulation is restricted to a subset of digital inputs (DI) and digital outputs (DO) relevant to the marking cycle. For each scenario, the Python application defines a mask specifying which bits are affected and the forced values (0/1). Although the architecture is prepared to support additional parameters—such as persistence, pulse patterns, duty cycle, or timing offsets—these capabilities were not required in the experiments conducted in this work.

The logic embedded in PLC 2 synchronizes the write operations issued by Python, applies the selected mask, and ensures that, when no attack is active, the real signals are forwarded without modification. This mechanism operates symmetrically for DI and DO: DI values originating from the digital twin and DO commands originating from PLC 1 can both be intercepted and modified in PLC 2 before being incorporated into the corresponding data flow—either toward PLC 1 via PROFINET I/O or toward the digital twin via S7 PUT/GET.

This design enables reproducing the effect of a Man-in-the-Middle attack on process data while preserving system stability and ensuring full reproducibility of the experimental scenarios. Figure 6 illustrates the internal operation for the DI-manipulation case, showing how PLC 2 applies the bit-mask logic to merge genuine inputs from the digital twin with the manipulated values supplied by the Python application before exposing the resulting data to PLC 1 via PROFINET I/O. This mechanism establishes the basis for the attack scenarios presented next, starting with the reconnaissance phase described in Section 4.4.

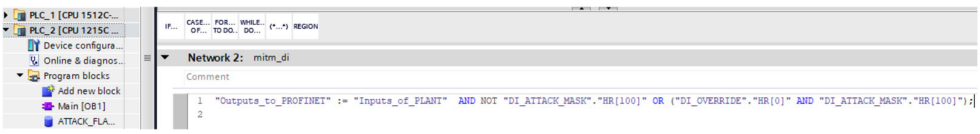


Figure 6. PLC 2-bit mask logic used for DI manipulation in the logical MitM attack.

4.4. Pre-Attack Reconnaissance for the MitM Attack

Before activating each manipulation scenario, the initial reconnaissance phase that an attacker would typically perform after gaining access to the testbed's OT network is reproduced. This phase enables the identification of the devices present in the industrial segment, their IP addresses, and the services they expose—without requiring intrusive techniques. The testbed faithfully replicates the information an adversary could obtain through passive traffic inspection or superficial device probing, providing a realistic starting point for the subsequent manipulation of PROFINET I/O signals.

4.4.1. Identification of Devices in the Industrial Network

Figure 7 shows the ARP/Ping scan executed from Kali Linux (nmap-sn), which identifies active devices on the 192.168.0.0/24 subnet. The analysis reveals several industrial components, including the Siemens S7-1200 PLC (192.168.0.2), which exposes the process-related digital signals via PROFINET I/O. This information allows selecting the target on which the RTC1-manipulation scenarios will be carried out in the following sections.

```
┌──(root㉿kali)-[~]
└─# sudo nmap -sn 192.168.0.0/24
Starting Nmap 7.95 ( https://nmap.org ) at 2026-02-05 11:15 EST
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or speci
fy valid servers with --dns-servers
Nmap scan report for 192.168.0.1
Host is up (0.00049s latency).
MAC Address: E0:DC:A0:8D:79:41 (Siemens Industrial Automation Products, Chengdu)
Nmap scan report for 192.168.0.2
Host is up (0.0036s latency).
MAC Address: E0:DC:A0:97:F9:4A (Siemens Industrial Automation Products, Chengdu)
Nmap scan report for 192.168.0.3
Host is up (0.0025s latency).
MAC Address: 4C:E7:05:16:D1:BA (Siemens Industrial Automation Products, Chengdu)
Nmap scan report for 192.168.0.100
Host is up (0.0051s latency).
MAC Address: 38:4B:24:84:01:9E (Siemens AG)
Nmap scan report for 192.168.0.253
Host is up (0.00023s latency).
MAC Address: 7C:57:58:D4:9E:37 (HP)
Nmap scan report for 192.168.0.150
Host is up.
Nmap done: 256 IP addresses (6 hosts up) scanned in 2.10 seconds
```

Figure 7. ARP/Ping scan showing active devices in the 192.168.0.0/24 subnet.

4.4.2. Discovery of Open Ports

After identifying the target device's IP address during the initial reconnaissance phase, the next step is to analyze the industrial services exposed by that device. For this purpose, a port scan is performed using Nmap from Kali Linux, focusing on TCP ports 102 and 502, associated with the S7comm (Siemens) and Modbus/TCP protocols, respectively. Figure 8 shows the results of the analysis, confirming that both services are accessible on the Siemens S7-1200 PLC (192.168.0.2). This validates its role as an IO Device and helps determine which communication channels may be manipulated in the subsequent attack scenarios.

```
┌──(root㉿kali)-[~]
└─# nmap -p- -sS --min-rate 5000 --open -vvv -n 192.168.0.2
Starting Nmap 7.95 ( https://nmap.org ) at 2026-02-05 10:44 EST
Initiating ARP Ping Scan at 10:44
Scanning 192.168.0.2 [1 port]
Completed ARP Ping Scan at 10:44, 0.09s elapsed (1 total hosts)
Initiating SYN Stealth Scan at 10:44
Scanning 192.168.0.2 [65535 ports]
Discovered open port 502/tcp on 192.168.0.2
Discovered open port 102/tcp on 192.168.0.2
Completed SYN Stealth Scan at 10:44, 30.01s elapsed (65535 total ports)
Nmap scan report for 192.168.0.2
Host is up, received arp-response (0.016s latency).
Scanned at 2026-02-05 10:44:20 EST for 30s
Not shown: 58021 closed tcp ports (reset), 7512 filtered tcp ports (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT    STATE SERVICE  REASON
102/tcp open  iso-tsap syn-ack ttl 30
502/tcp open  mbap     syn-ack ttl 30
MAC Address: E0:DC:A0:97:F9:4A (Siemens Industrial Automation Products, Chengdu)

Read data files from: /usr/share/nmap
Nmap done: 1 IP address (1 host up) scanned in 30.39 seconds
           Raw packets sent: 148463 (6.532MB) | Rcvd: 58025 (2.321MB)
```

Figure 8. Port scan showing open services on the device.

4.4.3. Fingerprinting of Exposed Industrial Services

After identifying the target host, a preliminary inspection of the accessible services is performed to determine the type of device and the industrial protocols enabled. The Siemens S7-1200 PLC located at 192.168.0.2 exposes the following relevant services:

- 102/TCP—S7comm (ISO TSAP): The service response confirms that the device is a genuine Siemens PLC and reveals hardware and firmware information (e.g., Module 6ES7 215-1BG40-0XB0, Firmware 4.3.1). This fingerprinting step confirms the device's identity and its role as a PROFINET I/O Device within the testbed.
- 502/TCP—Modbus TCP: Port associated with the MB_SERVER block running on PLC 2, which exposes the internal registers used by the Python application for DI/DO manipulation. Its presence reflects the hybrid nature of the testbed, where PROFINET and Modbus/TCP coexist for experimental purposes.

The detected MAC address—assigned to Siemens Industrial Automation Products (Chengdu)—further confirms the correspondence between the real PLC hardware and the information obtained during fingerprinting. Figure 9 shows the results of a version-detection scan (nmap-sCV), which identifies the active industrial protocols and provides detailed module and firmware information, relevant for characterizing the device's role within the testbed.



Figure 9. Service fingerprinting scan showing open ports on the device.

4.4.4. Relation to the Manipulation Scenarios

The information collected during the reconnaissance phase directly supports the MitM manipulations developed in the following sections. In particular:

- Identification of the PLC exposing key industrial services: The attacker determines which device publishes ports associated with process control (e.g., PROFINET, S7comm, or Modbus/TCP), allowing the Siemens S7-1200 PLC to be profiled as the primary target for signal manipulation.
- Detection of the Modbus/TCP service: The presence of the Modbus server (MB_SERVER) on PLC 2 enables the modification of the DI/DO values packaged in its internal registers, complementing the manipulation performed over the PROFINET I/O channel and providing the insertion point for the logical MitM mechanism.
- Confirmation of the S7comm service and the PLC's role: The exposure of port 102/TCP (S7comm) confirms the device's identity, its communication role within the PROFINET cycle, and the coherence between the fingerprinted services and the RTC1 traffic analyzed during the MitM scenarios.

Based on this information, scenarios A1–A2 are designed to demonstrate how selective manipulation of DI/DO signals transmitted through PROFINET I/O can induce incorrect behaviors in PLC 1 without producing visible inconsistencies at the physical layer of the process.

4.5. Experimental MitM Attack Scenarios

This section describes the attack scenarios implemented in the testbed. Each scenario specifies which signals are manipulated, under which conditions the attack is triggered, and the intended impact on PLC 1. The manipulation is carried out by the Python application, which alters a subset of the DI/DO signals exposed by PLC 2 through the Modbus/TCP interface before they are incorporated into the PROFINET I/O cyclic exchange.

4.5.1. Scenario A1—Manipulation of Sensors B1 and B2

This scenario represents an attacker who forces the DI bits corresponding to sensors B1 and B2 while still observing their real values. The manipulation alters the GEMMA-based logic in PLC 1 without generating PROFINET alarms, thereby implementing a realistic DI-level logical MitM attack.

- Objective: Force PLC 1 to advance in the automatic sequence despite the absence of a physical workpiece. The attacker simulates the sequential activation of B1 (workpiece at loading) and B2 (workpiece in the marking area), causing the process to run through loading, marking, and ejection without any real physical intervention.
- Initial State and Affected Signals: The system is in automatic mode (GEMMA F1), waiting for a workpiece at the loading position (see Figure 4). DI bits B1 and B2 are selected for manipulation.
- Mechanism of the MitM Attack: The Python application modifies the DI values of B1 and B2 in the MB_SERVER registers of PLC 2. This PLC then applies its internal bit-mask logic to replace the real DI values with the forged ones before assembling the RTC1 process-data payload that is sent to PLC 1 over PROFINET I/O.

Figure 10 illustrates an example of DI-bit manipulation affecting signals B1 and B2 in PLC 2 during the logical MitM attack.

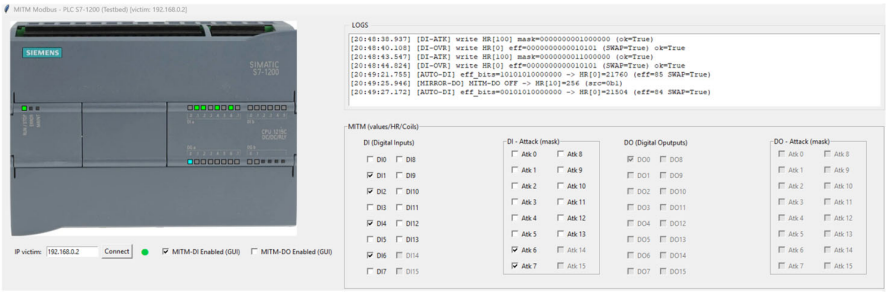


Figure 10. Example of DI-bit manipulation (B1 and B2) in PLC 2.

The effect of the manipulation appears immediately in the internal behavior of the system. Figure 11 shows the online view of PLC 2, where the real process value, the active attack mask, and the overridden value propagated to PLC 1 are displayed simultaneously. This confirms that the altered bits are generated inside PLC 2 and do not originate from the physical process.

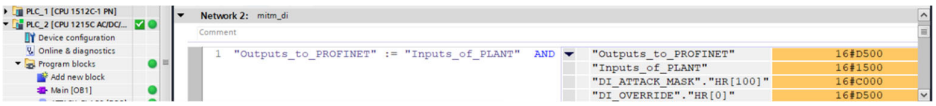


Figure 11. Online view of PLC 2 showing real and manipulated values.

- Effect: PLC 1 remains in GEMMA macro-state F1 (automatic mode) but proceeds incorrectly through the operational sequence (loading → marking → ejection) due to the forced DI bits (Figure 12). PLC 1 interprets the manipulated signals as genuine and executes the full cycle even though no physical workpiece exists.

- Evidence in PROFINET I/O traffic: Port-mirrored captures (Figure 13) show that the RTC1 frames contain the manipulated B1/B2 bits in the SDU, even though the digital twin did not generate these events. This confirms that the logical MitM attack modifies the SDU-level content while preserving protocol-level validity.

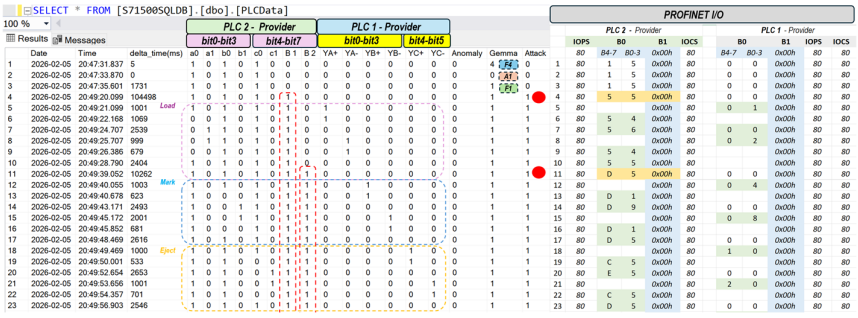


Figure 12. Correlation of SQL Server, GEMMA, and PROFINET data in scenario A1.

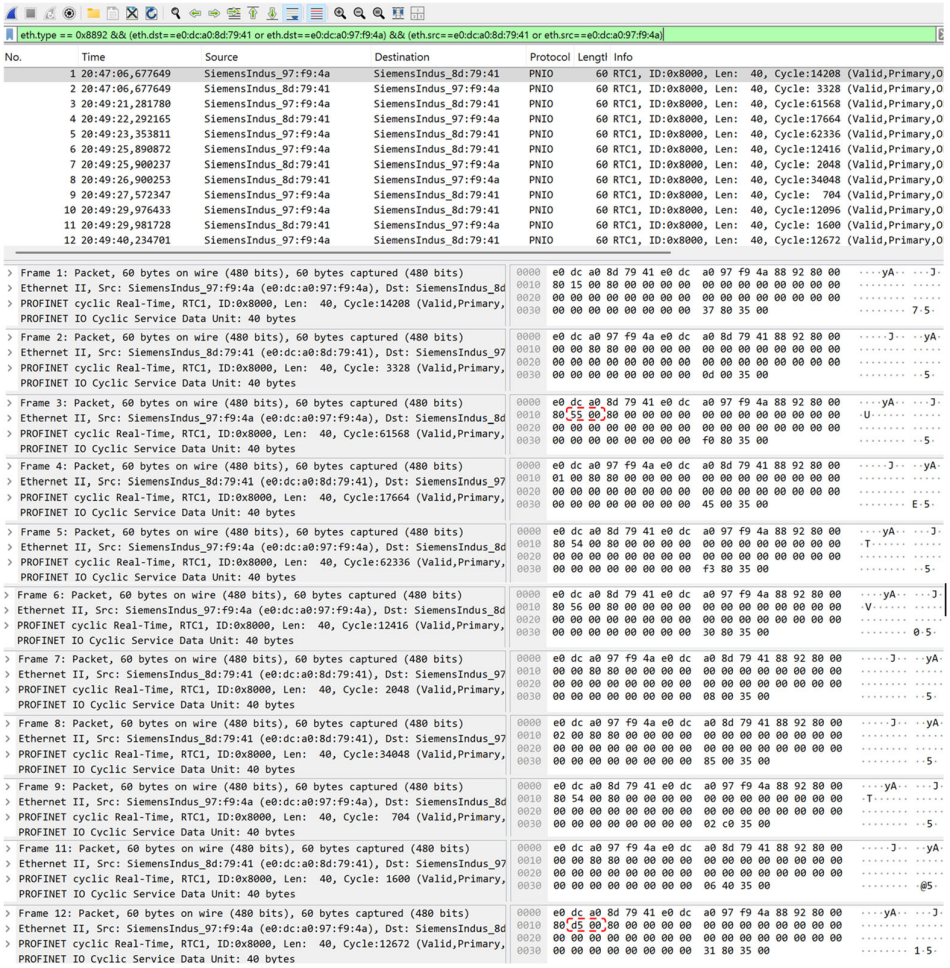


Figure 13. Wireshark capture of PROFINET RTC1 traffic in scenario A1.

Once the affected byte is identified, Figure 13 shows the corresponding forced bits in the RTC1 frames captured in Wireshark, offering clear packet-level evidence of the manipulation propagated to PLC 1.

4.5.2. Scenario A2—Manipulation of Actuator B (Marking Device)

In addition to scenario A1, which focuses on the manipulation of digital input (DI) signals by forcing selected bits, this scenario represents an attacker manipulating one of

the digital output bits responsible for actuating the bistable solenoid valve that governs the marking actuator. The attack is carried out by a malicious application through the Modbus/TCP interface, forcing a premature activation of the marking up command while PLC 1 continues issuing its legitimate command. By injecting a false value into the digital output prior to the actual process transition, the attack alters the mechanical behavior of the actuator without triggering PROFINET alarms, thereby implementing a realistic MitM attack at the digital output (DO) level.

- Objective: To prevent the marking operation from being correctly completed during the automatic sequence. By prematurely issuing a false command to the solenoid valve that triggers marking up, the attacker causes the valve to become blocked when the process later sends the legitimate marking down command, due to the reception of conflicting instructions.
- Initial State and Affected Signals: The system is operating in automatic mode (GEMMA F1), and the process is producing parts under normal conditions. Digital output DO 3, responsible for commanding the marking actuator to the up position, is selected for manipulation.
- MitM Attack Mechanism: While the process is running, the Python-based application writes a false value to the Modbus register corresponding to DO 3 = true (YB−) in PLC 2. This artificial activation is injected milliseconds before the legitimate command generated by PLC 1 during the transition to the marking stage (DO 2 = true (YB+). The premature activation interferes with the actual command, preventing the actuator from executing the marking operation and causing PLC 1 to generate a process fault due to an unexpected actuator response.

Figure 14 below shows the attacker interface injecting the malicious activation command through the aforementioned application.

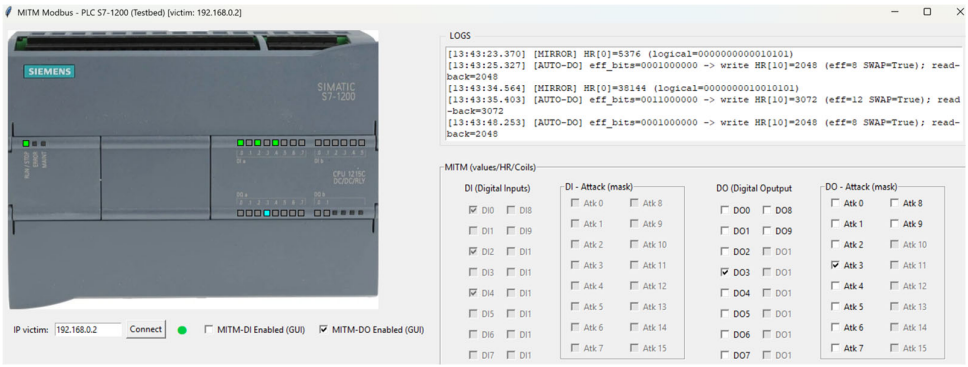


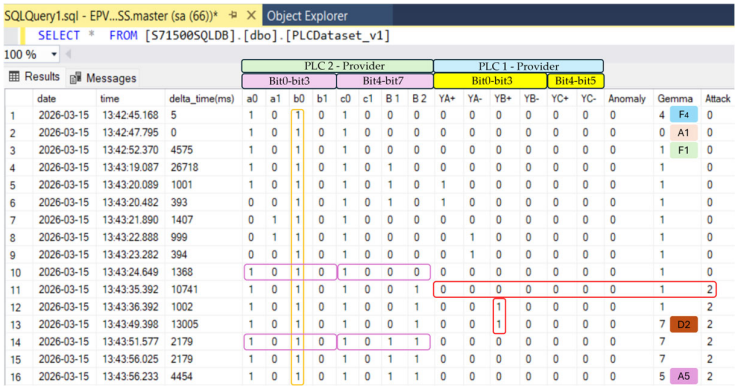
Figure 14. Manipulation of the DO signal YB− in PLC 2.

The outcome of the attack is recorded in the dataset stored in SQL Server, as well as in the PROFINET IO packet captures obtained using Wireshark, which are presented in Figures 15 and 16 below.

These figures, together with the previous one, demonstrate how the testbed captures all process events in real time. It simultaneously records the anomalous conditions generated in the process, the exact correspondence between each applied modification and its physical effect on the control sequence, the GEMMA operating mode in which each deviation occurs, and the type of detected attack (1 and 2 for DI and DO manipulation, respectively).

In this context, Figure 15 illustrates how these recorded variables evolve during the attack through the corresponding GEMMA sequence: F4 (system in manual mode) → A1 (system stopped in the initial state) → F1 (automatic normal production) → D2 (fault) → A5 (preparation for restart after a fault). To facilitate the interpretation of the attack,

particular attention is drawn to the state of sensor b0 (indicating the marker in the up position, with no change despite the issued command), the field Attack = 2 (indicating an output-level attack), and the marking command (YB+ = true) issued by PLC 1 to the solenoid valve controlling the marking unit during its automatic production sequence.



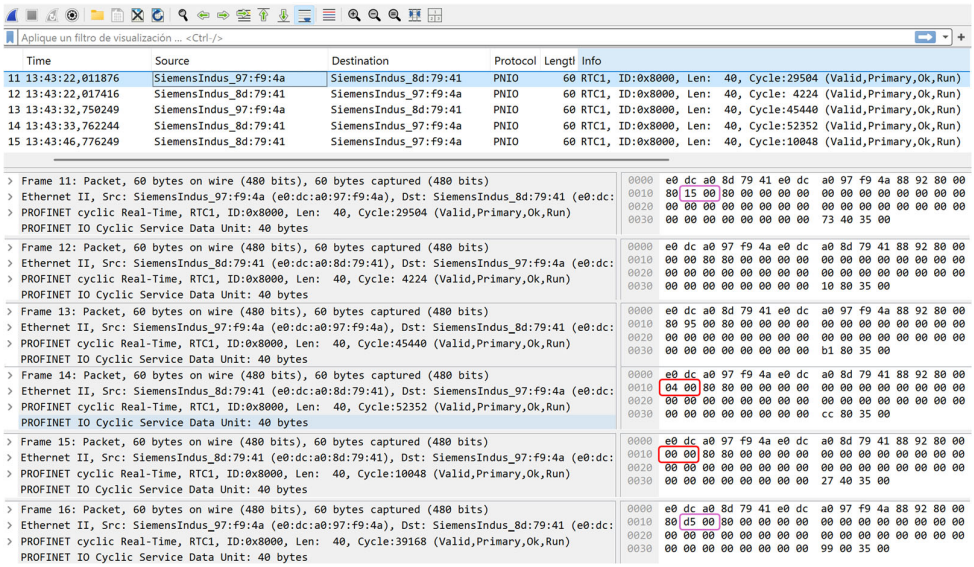
SQLQuery1.sql - EPV...SS.master (sa (66))* | Object Explorer

```
SELECT * FROM [S71500SQLDB].[dbo].[PLCDataset_v1]
```

100 % — Results — Messages

				PLC 2 - Provider							PLC 1 - Provider								
				Bit0-bit3				Bit4-bit7			Bit0-bit3				Bit4-bit5				
	date	time	delta_time(ms)	a0	a1	b0	b1	c0	c1	B 1	B 2	YA+	YA-	YB+	YB-	YC+	YC-	Anomaly	Gemma	Attack
1	2026-03-15	13:42:45.168	5	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	4 F4	0
2	2026-03-15	13:42:47.795	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0 A1	0
3	2026-03-15	13:42:52.370	4575	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1 F1	0
4	2026-03-15	13:43:19.087	26718	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	1	0
5	2026-03-15	13:43:20.089	1001	1	0	1	0	1	0	1	0	1	0	0	0	0	0	0	1	0
6	2026-03-15	13:43:20.482	393	0	0	1	0	1	0	1	0	1	0	0	0	0	0	0	1	0
7	2026-03-15	13:43:21.890	1407	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0
8	2026-03-15	13:43:22.888	999	0	1	1	0	1	0	0	0	0	1	0	0	0	0	0	1	0
9	2026-03-15	13:43:23.282	394	0	0	1	0	1	0	0	0	0	1	0	0	0	0	0	1	0
10	2026-03-15	13:43:24.649	1368	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0
11	2026-03-15	13:43:35.392	10741	1	0	1	0	1	0	0	1	0	0	0	0	0	0	0	1	2
12	2026-03-15	13:43:36.392	1002	1	0	1	0	1	0	0	1	0	0	1	0	0	0	0	1	2
13	2026-03-15	13:43:49.398	13005	1	0	1	0	1	0	0	1	0	0	1	0	0	0	0	7 D2	2
14	2026-03-15	13:43:51.577	2179	1	0	1	0	1	0	1	1	0	0	0	0	0	0	0	7	2
15	2026-03-15	13:43:56.025	2179	1	0	1	0	1	0	1	1	0	0	0	0	0	0	0	7	2
16	2026-03-15	13:43:56.233	4454	1	0	1	0	1	0	1	1	0	0	0	0	0	0	0	5 A5	2

Figure 15. Correlation of SQL Server, GEMMA, and PROFINET data in scenario A2.



Aplique un filtro de visualización ... <Ctrl-/>

	Time	Source	Destination	Protocol	Lengtl	Info
11	13:43:22,011876	SiemensIndus_97:f9:4a	SiemensIndus_8d:79:41	PNIO	60	RTC1, ID:0x8000, Len: 40, Cycle:29504 (Valid,Primary,Ok,Run)
12	13:43:22,017416	SiemensIndus_8d:79:41	SiemensIndus_97:f9:4a	PNIO	60	RTC1, ID:0x8000, Len: 40, Cycle: 4224 (Valid,Primary,Ok,Run)
13	13:43:32,750249	SiemensIndus_97:f9:4a	SiemensIndus_8d:79:41	PNIO	60	RTC1, ID:0x8000, Len: 40, Cycle:45440 (Valid,Primary,Ok,Run)
14	13:43:33,762244	SiemensIndus_8d:79:41	SiemensIndus_97:f9:4a	PNIO	60	RTC1, ID:0x8000, Len: 40, Cycle:52352 (Valid,Primary,Ok,Run)
15	13:43:46,776249	SiemensIndus_8d:79:41	SiemensIndus_97:f9:4a	PNIO	60	RTC1, ID:0x8000, Len: 40, Cycle:10048 (Valid,Primary,Ok,Run)

```
Frame 11: Packet, 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
Ethernet II, Src: SiemensIndus_97:f9:4a (e0:dc:a0:97:f9:4a), Dst: SiemensIndus_8d:79:41 (e0:dc:
PROFINET cyclic Real-Time, RTC1, ID:0x8000, Len: 40, Cycle:29504 (Valid,Primary,Ok,Run)
PROFINET IO Cyclic Service Data Unit: 40 bytes
0000  e0 dc a0 8d 79 41 e0 dc  a0 97 f9 4a 88 92 80 00
0010  80 15 00 80 00 00 00 00  00 00 00 00 00 00 00 00
0020  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0030  00 00 00 00 00 00 00 00  73 40 35 00

Frame 12: Packet, 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
Ethernet II, Src: SiemensIndus_8d:79:41 (e0:dc:a0:8d:79:41), Dst: SiemensIndus_97:f9:4a (e0:dc:
PROFINET cyclic Real-Time, RTC1, ID:0x8000, Len: 40, Cycle: 4224 (Valid,Primary,Ok,Run)
PROFINET IO Cyclic Service Data Unit: 40 bytes
0000  e0 dc a0 97 f9 4a e0 dc  a0 8d 79 41 88 92 80 00
0010  00 00 80 80 00 00 00 00  00 00 00 00 00 00 00 00
0020  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0030  00 00 00 00 00 00 00 00  10 80 35 00

Frame 13: Packet, 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
Ethernet II, Src: SiemensIndus_97:f9:4a (e0:dc:a0:97:f9:4a), Dst: SiemensIndus_8d:79:41 (e0:dc:
PROFINET cyclic Real-Time, RTC1, ID:0x8000, Len: 40, Cycle:45440 (Valid,Primary,Ok,Run)
PROFINET IO Cyclic Service Data Unit: 40 bytes
0000  e0 dc a0 8d 79 41 e0 dc  a0 97 f9 4a 88 92 80 00
0010  80 95 00 80 00 00 00 00  00 00 00 00 00 00 00 00
0020  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0030  00 00 00 00 00 00 00 00  b1 80 35 00

Frame 14: Packet, 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
Ethernet II, Src: SiemensIndus_8d:79:41 (e0:dc:a0:8d:79:41), Dst: SiemensIndus_97:f9:4a (e0:dc:
PROFINET cyclic Real-Time, RTC1, ID:0x8000, Len: 40, Cycle:52352 (Valid,Primary,Ok,Run)
PROFINET IO Cyclic Service Data Unit: 40 bytes
0000  e0 dc a0 97 f9 4a e0 dc  a0 8d 79 41 88 92 80 00
0010  04 00 80 80 00 00 00 00  00 00 00 00 00 00 00 00
0020  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0030  00 00 00 00 00 00 00 00  cc 80 35 00

Frame 15: Packet, 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
Ethernet II, Src: SiemensIndus_8d:79:41 (e0:dc:a0:8d:79:41), Dst: SiemensIndus_97:f9:4a (e0:dc:
PROFINET cyclic Real-Time, RTC1, ID:0x8000, Len: 40, Cycle:10048 (Valid,Primary,Ok,Run)
PROFINET IO Cyclic Service Data Unit: 40 bytes
0000  e0 dc a0 97 f9 4a e0 dc  a0 8d 79 41 88 92 80 00
0010  00 00 80 80 00 00 00 00  00 00 00 00 00 00 00 00
0020  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0030  00 00 00 00 00 00 00 00  27 40 35 00

Frame 16: Packet, 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
Ethernet II, Src: SiemensIndus_97:f9:4a (e0:dc:a0:97:f9:4a), Dst: SiemensIndus_8d:79:41 (e0:dc:
PROFINET cyclic Real-Time, RTC1, ID:0x8000, Len: 40, Cycle:39168 (Valid,Primary,Ok,Run)
PROFINET IO Cyclic Service Data Unit: 40 bytes
0000  e0 dc a0 8d 79 41 e0 dc  a0 97 f9 4a 88 92 80 00
0010  80 d5 00 80 00 00 00 00  00 00 00 00 00 00 00 00
0020  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0030  00 00 00 00 00 00 00 00  99 00 35 00
```

Figure 16. Wireshark capture of PROFINET RTC1 frames during scenario A2.

The fact that PLC 1 does not receive positive feedback within a specified time after issuing the command to the plant (marker down) causes the controller program itself to trigger a transition to state D2, thereby indicating to the operator a fault in actuator B (marker). However, no actual fault exists; rather, the solenoid valve is blocked as a consequence of the attack. This condition is not known to PLC 1.

Figure 16 shows a Wireshark capture of PROFINET IO RTC1 cyclic frames, which is particularly useful for identifying SDU-level manipulation consistent with a MitM attack on digital outputs. It illustrates both the command issued by PLC 1 to initiate the downward movement of the marking actuator and the corresponding state of the plant sensors before and after the attack.

By analyzing the sensor data transmitted by the plant, it can be observed that sensor b0 of actuator B does not change state, despite PLC 1 having issued the command YB+ = true (04–00 hex = 00000100; see Figure 15).

This behavior is a direct consequence of the attack, in which a forged digital output is injected into the plant, resulting in a malfunction of actuator B (marker).

Likewise, the testbed supports combined DI/DO manipulation, in which both masks are applied simultaneously to the SDU and the plant. While this variant represents a more aggressive attack configuration, it does not introduce a fundamentally different mechanism from those described in Scenarios A1 and A2, as it corresponds to a combination of input- and output-level manipulations.

4.6. Validation of Communication Integrity (Ping and PROFINET Timing)

This section evaluates whether the execution of the MitM attacks affects the stability of the communication channel. Since the proposed mechanism modifies only the RTC1 SDU content, it is essential to confirm that the underlying communication layers remain unaffected. Four conditions were analyzed:

- Baseline: No MitM module active.
- Attack-Idle: MitM module active but without manipulating SDU content.
- Attack-DI: Manipulation of the digital input (DI) byte (scenario A1).
- Attack-DO: Manipulation of the digital output (DO) byte (scenario A2).

The assessment comprises three complementary measurements: ICMP round-trip time (RTT), PROFINET RTC1 cycle timing derived from Wireshark captures, and throughput evaluation. Together, these measurements address RQ2 by determining whether the attack introduces measurable deviations compared with baseline behavior.

4.6.1. ICMP Round-Trip Time Analysis

ICMP echo requests were used to assess whether enabling the MitM module—or performing SDU-level manipulation—induces measurable changes in L3 connectivity. Two communication paths were evaluated: EW → PLC 2 and DT → PLC 1, using 250 ICMP probes per condition. As summarized in Table 4, both paths exhibited zero packet loss across all four scenarios (baseline, idle, DI, and DO).

Table 4. ICMP RTT statistics for the four experimental conditions.

Condition	Direction	Mean (ms)	SD (ms)	Min (ms)	Max (ms)	Loss (%)
Baseline	DT→PLC1	1.027	0.162	1	2	0.0
Baseline	EW→PLC2	4.100	2.163	1	12	0.0
Attack-Idle	DT→PLC1	1.014	0.116	1	2	0.0
Attack-Idle	EW→PLC2	4.508	2.33	1	13	0.0
Attack-DI	DT→PLC1	1.028	0.165	1	2	0.0
Attack-DI	EW→PLC2	4.624	2.196	1	12	0.0
Attack-DO	DT→PLC1	1.013	0.112	1	2	0.0
Attack-DO	EW→PLC2	4.496	2.374	1	12	0.0

RTT measurements for the DT → PLC 1 direction remained tightly concentrated around their nominal values, with only minor variations attributable to normal switched-Ethernet jitter. These results indicate that neither enabling the MitM module nor performing SDU manipulation introduces additional L3 latency on this path.

In contrast, the EW → PLC 2 direction presented higher dispersion in RTT values, a behavior consistently observed across all conditions and reflecting the inherent variability of this route rather than any attack-induced effects. This wider spread is also consistent with host-side effects on the Engineering Workstation, where TIA Portal and the MSSQL server operate concurrently, introducing CPU load and kernel-level scheduling delays that affect ICMP timestamping. Such effects are well-known to increase RTT variability without indicating any degradation in network-layer performance or any influence from the MitM mechanism. Importantly, the differences between baseline, idle, and DI/DO

scenarios remained within normal operational fluctuations, and no systematic latency increase was observed.

Figure 17 consolidates these findings by displaying the RTT mean and standard deviation for both communication paths. The figure highlights the stability of the DT → PLC 1 path and the modest, non-systematic variability of the EW → PLC 2 path. Overall, the ICMP service remains stable under all scenarios, confirming that the MitM module operates exclusively at the SDU level without impacting network-layer connectivity, jitter, or packet delivery.

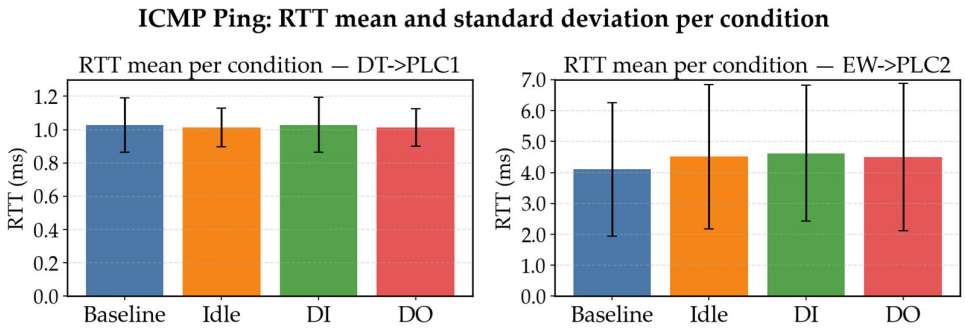


Figure 17. ICMP RTT measurements for the four experimental conditions.

4.6.2. PROFINET I/O Cycle Timing Analysis

RTC1 PROFINET traffic was analyzed to determine whether enabling the MitM module—or performing SDU-level manipulation—introduces timing disturbances in the cyclic I/O exchange. For each experimental condition, all RTC1 frames captured through port mirroring were extracted, and their inter-arrival times were used to estimate the effective cycle time.

As summarized in Table 5, the mean cycle time remained effectively constant across all scenarios, with no systematic deviation from the baseline. Jitter values were likewise comparable between conditions and consistent with the timestamp variability expected when captures are performed on a non-real-time operating system. Minimum inter-arrival times were bounded by timestamp resolution. In contrast, occasional maximum spikes appeared in all scenarios—including the baseline—indicating that such variations originate from measurement-side jitter rather than from the MitM activity.

Table 5. PROFINET RTC1 cycle-time statistics across the four experimental conditions.

Condition	Frames	Mean (ms)	SD (ms)	Min (ms)	Max (ms)
Baseline	351,972	0.5	0.728	0.0	3.147
Attack-Idle	387,239	0.5	0.782	0.0	3.363
Attack-DI	488,238	0.5	0.782	0.0	3.346
Attack-DO	419,406	0.5	0.834	0.0	3.417

Overall, the results show that neither enabling the MitM module nor performing DI-based or DO-based SDU manipulation affects PROFINET RTC1 cycle timing or its real-time scheduling behavior. The cyclic I/O exchange remains stable and deterministic under all tested conditions.

The inter-arrival times obtained from the pooled PLC 1 ↔ PLC 2 capture are also consistent with the packet-rate measurements observed in Wireshark. During the mirrored-port acquisition, the traffic reached approximately 2 kpps, reflecting the combined RTC1 exchanges of both controllers (≈1000 packets/s per PLC). Because the inter-arrival time corresponds to the inverse of the packet rate, a throughput of 2000 packets/s yields an

expected mean interval of 1/2000 s (≈0.5 ms), in agreement with the distribution shown in Figure 18. Occasional jitter produces isolated inter-arrival values exceeding 1 ms, explaining the extended whiskers in the boxplot while having no effect on the mean value.

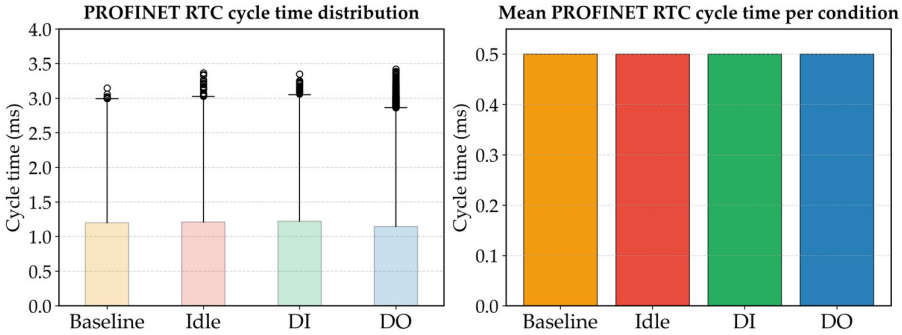


Figure 18. PROFINET RTC1 cycle-time distribution for PLC 1 ↔ PLC 2 traffic.

4.6.3. Throughput Evaluation

Throughput was evaluated by analyzing the RTC1 packet rate (PPS) under the four experimental conditions. As summarized in Table 6, the baseline exhibits a stable packet-exchange pattern with an average throughput of approximately 4 kpps. The attack-idle condition presents the largest deviation relative to the baseline, whereas the DI- and DO-based manipulation scenarios introduce only marginal differences. Across all attack modes, throughput variations remain within ±0.3% for both PPS and Bytes/s, and the increased variability observed in the idle and DO scenarios reflects normal second-level binning effects and the micro-timing introduced by the MITM relay.

Table 6. PROFINET throughput statistics for the four experimental conditions.

Scenario	PPS Mean	PPS SD	kB/s Mean	kB/s SD	ΔPPS (%)
Baseline	4073.01	307.47	248.30	18.74	0.00%
Attack-Idle	4068.86	302.12	248.10	18.42	−0.10%
Attack-DI	4082.76	166.56	248.93	10.15	0.24%
Attack-DO	4072.78	282.94	248.34	17.25	−0.01%

These deviations are modest and compatible with the expected behavior of a lightly loaded PROFINET RT network. Importantly, this throughput metric aggregates all RTC1 frames counted per second on the mirrored port, in contrast to the pooled RTC1 inter-arrival analysis in Section 4.6.2, which reflects an effective rate of ≈2 kpps. The observed differences, therefore, represent measurement-scope effects rather than protocol-level disturbances.

Numerically, the Idle and DO scenarios exhibit PPS values marginally below the baseline (ΔPPS = −0.10% and −0.01%, respectively), whereas the DI scenario is slightly above it (+0.24%). These shifts are two orders of magnitude smaller than the within-scenario dispersion (SD ≈ 283–307 PPS) and fall within the normal jitter of lightly loaded PROFINET RT networks. Scenarios with reduced process activity—such as Idle and DO—are inherently more sensitive to timestamp granularity, mirrored-port timing, and second-level binning effects, which can slightly lower the PPS count in individual one-second measurement windows. Conversely, the higher DI activity increases the probability of capturing additional RTC1 frames within the same interval. As similar fluctuations also appear in the baseline condition, these differences cannot be attributed to the logical MitM manipulation, but rather to measurement-side variability.

Overall, the measurements confirm that neither enabling the MITM module nor manipulating DI or DO values compromises RTC1 timing, protocol validity, or network

load. Although throughput analysis is primarily employed here to confirm the execution of the attack and verify the absence of traffic-level side effects, the testbed also supports future research on automated anomaly and attack detection in PROFINET I/O systems using AI-based methods—an area that remains largely unexplored.

5. Results

5.1. Scenario A1—Manipulation of DI Signals

Scenario A1 validates the capability of the MitM module to selectively manipulate the PROFINET I/O exchange between PLC 2 and PLC 1, altering the SDU content delivered to the controller without affecting the timing or validity of the cyclic RTC1 frames. The RTC1 traffic captured through port mirroring confirms that the attacker successfully modifies the SDU bytes corresponding to signals B1 and B2, which appear as active despite the absence of any physical events in the digital twin. These injected values are embedded in otherwise valid RTC1 frames—preserving structural integrity, temporal coherence, and correct IOPS/IOCS indicators—thereby preventing detection at the communication layer. Figure 13 illustrates the modified SDU and its seamless integration within the cyclic exchange.

The dataset recorded in MSSQL (Figure 12) provides a synchronized view of the physical process state, the PROFINET-delivered values, the associated GEMMA transitions, and the timing of the attack activation. Throughout the manipulation interval, the forged bits remain continuously active, causing PLC 1 to infer the presence of a workpiece even though no physical sensor asserts that condition. This confirms the consistency between the modified network traffic and the values logged at the application layer, demonstrating that the attack yields a logically valid yet physically incorrect process state.

Inspection of the control logic in TIA Portal (Figure 11) shows that PLC 1 integrates the received PROFINET digital inputs directly into its execution cycle without applying additional validation mechanisms. As a result, the forged input values propagate through the GEMMA state machine and the control chain without triggering alarms or fault conditions. From the controller's perspective, the manipulated SDU represents a legitimate and temporally coherent state transition.

5.2. Scenario A2—Manipulation of DO Signals

Scenario A2 confirms that the attacker can manipulate the digital-output layer of the control chain without altering the PROFINET SDU exchanged between PLC 1 and PLC 2. The RTC1 cyclic communication remains valid and unmodified, as the attacker neither intercepts nor rewrites the DO bytes transmitted from PLC 1. Instead, the forged YB− command is injected directly into the process image of PLC 2 via the Modbus/TCP interface. As a result, PLC 2 sends a falsified actuator command to the plant while still receiving an intact and legitimate SDU from PLC 1. Because the PROFINET frames are not altered in transit, Wireshark does not capture any SDU-level modification—only the physical consequence of the tampering on the actuator state. Figures 14–16 show the malicious command injection, its impact on the RTC1 SDU, and the resulting actuator behavior.

The dataset recorded in MSSQL captures the resulting inconsistency between the legitimate YB+ command issued by PLC 1 and the unchanged sensor feedback b0, which remains active despite the expected movement. This mismatch forces the controller to enter state D2 in the GEMMA sequence, reporting a marking-actuator fault even though no physical malfunction exists. The SQL log and the PROFINET captures exhibit perfect temporal alignment between the injected DO value (applied internally in PLC 2) and the anomalous process state, confirming that the testbed accurately records both the physical and logical consequences of the attack.

These results demonstrate that output-level manipulation can disrupt the control cycle without modifying the PROFINET SDU or triggering protocol-level alarms, thereby complementing the findings reported for Scenario A1.

5.3. Combined Interpretation of A1 and A2

These observations align with the communication-level measurements presented in Section 4.6. The MitM manipulation does not affect L3 latency or packet delivery (Section 4.6.1) and does not disturb the RTC1 cycle timing (Section 4.6.2). The throughput analysis (Section 4.6.3) shows only small variations—within ±0.3%—consistent with normal binning effects and the micro-timing introduced by the relay, none of which interfere with the cyclic I/O exchange. Taken together, the results demonstrate that the attacker can modify the logical perception of the process without altering PROFINET timing behavior or violating protocol-level validity checks.

Overall, the hybrid testbed enables simultaneous observation of the physical domain—represented by the digital twin—and the logical domain through PROFINET captures and MSSQL Server logging. This integrated perspective provides clear evidence of the effectiveness of the attack. It highlights the need for complementary validation mechanisms in industrial systems, as SDU-level manipulation can remain invisible to both the controller and the operator while still inducing incorrect process execution.

The impact of the MitM module becomes most evident at the process-execution level, where each attack mode (idle, DI, DO) interacts differently with the behavior of the marking system. Although the idle mode does not alter the semantic execution of the process, it represents the stage in which the attacker is connected to the communication path while maintaining fully valid PROFINET cyclic traffic. In contrast, the DI and DO manipulation modes introduce changes at the application layer, despite the cyclic communication remaining structurally and temporally valid. Across all scenarios, RTC1 frames preserve their timing and maintain consistent IOPS/IOCS indicators, preventing the controller from detecting manipulation at the communication layer. However, only the DI and DO modes modify the logical behavior of the process—either by altering perceived sensor states (DI manipulation) or by forcing actuator commands (DO manipulation)—while remaining indistinguishable from valid RTC1 cyclic traffic.

6. Discussion

The results of scenario A1 demonstrate that selectively modifying the SDU in RTC1 frames enables an attacker to introduce logically valid yet physically incorrect process states without producing measurable deviations in the evaluated PROFINET cyclic exchange indicators. This behavior is consistent with the PROFINET I/O specification [24], in which the validity of a telegram depends on the structural integrity of IO Data Objects, the correctness of offsets, and valid IOPS/IOCS indicators, rather than on any semantic verification of payload meaning. As long as the frame structure and timing remain correct, the PLC accepts the modified values as legitimate inputs.

The experimental evidence confirms that coherent SDU manipulation is sufficient to influence process behavior without affecting traffic availability or PROFINET timing parameters. Such manipulation enables the attacker to force incorrect GEMMA transitions or generate fictitious sensor activations while maintaining valid cyclic communication, thereby making the manipulation undetectable by standard PROFINET RTC1 integrity mechanisms. This highlights a critical vulnerability: the controller's perception of the process can be altered while all network-level indicators remain nominal.

Scenario A2 complements these findings by demonstrating that output-level manipulation can disrupt the physical behavior of the process without modifying the PROFINET

SDU exchanged between PLC 1 and PLC 2. Because the attacker injects the forged actuator command directly into the PLC 2 process image, the RTC1 frames observed on the network remain structurally valid, temporally coherent, and indistinguishable from legitimate cyclic traffic. This makes the attack invisible to communication-layer monitoring while still forcing deterministic deviations in the actuator sequence, leading PLC 1 to misinterpret the plant state and transition to GEMMA fault mode (D2). The contrast between unmodified network traffic and incorrect actuator behavior further illustrates the limitations of relying solely on PROFINET cyclic-integrity indicators for anomaly detection.

The scenario also underscores the importance of reducing the attack surface in industrial networks. In the testbed, the exposure of TCP port 502—associated with a Modbus/TCP service under attacker control—can be identified through basic network-scanning techniques. Although the MitM attack presented here does not rely on this service, its presence illustrates how unnecessary services or misconfigurations may facilitate reconnaissance in real deployments. Measures such as disabling unused ports, enforcing network segmentation, and restricting administrative services are essential to minimize intrusion opportunities.

A key contribution of this work is the integration of three complementary data sources: the physical state from the digital twin, PROFINET traffic captured via port mirroring, and structured logs stored in MSSQL. This combination provides full traceability of the attack, from its manifestation at the communication layer to its propagation through the PLC logic. The ability to correlate physical behavior, network traffic, and control execution offers a comprehensive understanding of the consequences of SDU-level manipulation and strengthens the reliability of the analysis.

This study, however, presents certain limitations. The analysis targets a limited set of digital signals, a single attack configuration, and exclusively RTC1 traffic. More complex architectures, heterogeneous devices, higher-performance modes such as RTC3, and additional manipulation strategies remain unexplored. Likewise, the testbed does not evaluate redundancy mechanisms, PROFIsafe, or integrity layers that may affect the detectability of SDU manipulation.

Despite these limitations, the results show that semantic manipulation of the SDU constitutes a relevant threat in PROFINET I/O systems. The hybrid testbed demonstrates how SDU-level attacks can be introduced, how they propagate through the control logic, and how they remain unnoticed by routine RTC1 structural-integrity checks. This provides a solid foundation for future research on detection methods, semantic-validation techniques, and mitigation strategies against MitM attacks in industrial environments.

Regarding the research questions posed in Section 1.2, the experimental results confirm that (RQ1) the proposed hybrid testbed provides the flexibility required to reproduce realistic PROFINET I/O communication and safely execute controlled MitM attacks; (RQ2) such attacks do not introduce measurable deviations in RTT, throughput, or RTC1 cycle timing and therefore remain stealthy with respect to these communication-layer indicators at the communication layer; and (RQ3) the testbed is capable of generating reproducible baseline and attack datasets suitable for analysis, anomaly-detection research, and training activities in industrial cybersecurity.

Taken together, these insights articulate the broader implications of SDU-level semantic manipulation, clarify the practical limitations of the current implementation, and outline the conditions under which the proposed approach can be generalized to other communication architectures and industrial-control settings.

7. Conclusions and Future Work

This work demonstrates that a Man-in-the-Middle adversary intercepting RTC1 traffic between two PROFINET controllers can selectively modify process semantics without

altering cyclic-exchange timing or triggering protocol-level validity checks. Scenario A1 shows that injecting altered digital-input (DI) values induces incorrect GEMMA transitions in the consuming PLC, generating logically valid states that do not correspond to any real physical condition. Scenario A2 further confirms that manipulating actuator-command signals at the digital-output (DO) level can force deterministic deviations in the execution of the process without modifying the PROFINET SDU exchanged between PLC 1 and PLC 2. In both scenarios, the attacks preserve the structural integrity of the RTC1 frames—including IO Data Object organization and IOPS/IOCS semantics—rendering the manipulation indistinguishable from normal cyclic communication.

The combined use of a hybrid digital-twin-based testbed, PROFINET traffic captured via port mirroring, and structured MSSQL-based logging enabled detailed correlation between the physical state of the process, the logical perception of the controller, and the real-time manipulation introduced by the MitM module. This integration provides full traceability of the attack as it propagates from the communication layer to the control logic and offers a comprehensive, non-invasive platform for studying semantic manipulation attacks in PROFINET I/O systems. The methodology is reproducible and compatible with industrial environments where high-fidelity experimentation must not disrupt operational hardware.

As future work, the validated testbed will be used for the systematic generation of datasets aimed at advancing research in industrial cybersecurity for PROFINET-based systems. The infrastructure developed in this study supports the controlled reproduction of diverse SDU manipulation patterns. It enables the creation of labeled datasets that integrate physical (digital twin), logical (PROFINET RTC1/RTC3), and control-level (GEMMA and MSSQL logs) information. These datasets will serve as a foundation for evaluating anomaly-detection mechanisms based on machine-learning and advanced sequential-analysis methods. Beyond demonstrating the feasibility of both DI- and DO-level attacks, the proposed testbed opens new research avenues focused on the detection, analysis, and mitigation of Man-in-the-Middle threats in industrial cyber-physical systems.

These results consolidate the relevance of the proposed testbed as a scientific instrument for analyzing semantic-level attacks in industrial communication systems. They demonstrate its value as a foundation for generating high-quality industrial datasets and for supporting the future development of intelligent detection, validation, and protection mechanisms for PROFINET-based industrial cyber-physical systems.

Author Contributions: Conceptualization, J.V.M.-F.; methodology, J.V.M.-F.; software, J.V.M.-F.; validation, J.E.S.G., N.B., and Á.H.; formal analysis, J.V.M.-F.; investigation, J.V.M.-F.; data curation, J.V.M.-F.; writing—original draft preparation, J.V.M.-F.; writing—review and editing, J.V.M.-F., J.E.S.G., N.B., and Á.H.; visualization, J.V.M.-F.; supervision, J.E.S.G., N.B., and Á.H.; funding acquisition, J.E.S.G. and Á.H. All authors have read and agreed to the published version of the manuscript.

Funding: This publication is part of the AI4SECIoT project ("Artificial Intelligence for Securing IoT Devices"–C032.23), funded by the National Cybersecurity Institute (INCIBE), derived from a collaboration agreement signed between the National Institute of Cybersecurity (INCIBE) and the University of Burgos. This initiative is carried out within the framework of the Recovery, Transformation and Resilience Plan funds, financed by the European Union (Next Generation), the project of the Government of Spain that outlines the roadmap for the modernization of the Spanish economy, the recovery of economic growth and job creation, for solid, inclusive and resilient economic reconstruction after the COVID19 crisis, and to respond to the challenges of the next decade.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

AR	Application Relation (PROFINET)
ARP	Address Resolution Protocol
CER	Critical Entities Resilience Directive
CPS	Cyber–Physical System
CRA	Cyber Resilience Act
DCP	PROFINET Discovery and Configuration Protocol
DI	Digital Input
DO	Digital Output
DoS	Denial of Service
DT	Digital Twin
EAP-TLS	Extensible Authentication Protocol—Transport Layer Security
EW	Engineering Workstation
GEMMA	Guide for the Study of Machine Operating Modes
HMI	Human–Machine Interface
ICMP	Internet Control Message Protocol
ICPS	Industrial Cyber Physical System
ICS	Industrial Control System
ICT	Information and Communication Technologies
iDevice	PROFINET I/O Device mode (Siemens PLC)
IEC 62443	Industrial Cybersecurity Standards (IEC 62443 series)
I/O	Input/Output
IOCS	Input Output Consumer Status
IOPS	Input Output Provider Status
IP	Internet Protocol
ISO	International Organization for Standardization
IT	Information Technology
kB/s	Kilobytes per second
L3	Layer 3 (Network Layer)
MAC	Media Access Control (Ethernet hardware address)
MitM	Man in the Middle
MSSQL	Microsoft SQL Server
Modbus TCP	Modbus Transmission Control Protocol
NIS2	Network and Information Security Directive 2
OB1	Organization Block 1 (main PLC execution cycle)
OSI	Open Systems Interconnection
OT	Operational Technology
PAC	Programmable Automation Controller
PLC	Programmable Logic Controller
PROFIBUS	Process Field Bus
PROFINET I/O	PROcess Field NETwork Input/Output
PROFIsafe	PROFIBUS/PROFINET Safety Communication Profile
PUT/GET	Siemens S7 read/write communication service
PPS	Packets Per Second
RTC1	Real Time Class 1 (cyclic real-time communication)

RTC3	Real Time Class 3 (Isochronous Real Time)
RTT	Round Trip Time
SCADA	Supervisory Control and Data Acquisition
SD	Standard Deviation
SDU	Service Data Unit
S7comm	Siemens S7 communication protocol
SSMS	SQL Server Management Studio
SYN	TCP Synchronize flag (used to initiate a connection)
TCP	Transmission Control Protocol
TIA Portal	Totally Integrated Automation Portal
TSAP	Transport Service Access Point
UDP	User Datagram Protocol
VPN	Virtual Private Network

References

1. Serpanos, D. The Cyber-Physical Systems Revolution. *Computer* **2018**, *51*, 70–73. [CrossRef]
2. Javid, M.; Haleem, A.; Suman, R. Digital Twin Applications toward Industry 4.0: A Review. *Cogn. Robot.* **2023**, *3*, 71–92. [CrossRef]
3. Stranahan, J.; Soni, T.; Heydari, V. Supervisory Control and Data Acquisition Testbed Vulnerabilities and Attacks. In *Proceedings of the IEEE SoutheastCon, Huntsville, AL, USA, 11–14 April 2019*; IEEE: Piscataway, NJ, USA, 2019; pp. 1–6. [CrossRef]
4. Sinha, A.; Patel, S.S.; Kumar, A.; Vyas, O.P. Exploiting Vulnerabilities in the SCADA Modbus Protocol: An ICT Reliant Perspective. In *Advanced Network Technologies and Intelligent Computing*; Woungang, I., Dhurandher, S.K., Pattanaik, K.K., Verma, A., Verma, P., Eds.; Communications in Computer and Information Science; Springer: Cham, Switzerland, 2022; Volume 1534, pp. 94–108. [CrossRef]
5. Plager, A.; Olexa, E.; Gardner, D.; Torres, B.; Hays, A.; Farraj, A. A Study of SCADA System Vulnerabilities and Man-in-the-Middle Threats in Substation Operations. In *Proceedings of the 2025 IEEE Texas Power and Energy Conference (TPEC), National Harbor, MD, USA, 10–11 February 2025*; IEEE: Piscataway, NJ, USA, 2025; pp. 1–5. [CrossRef]
6. Qian, J.; Du, X.; Chen, B.; Qu, B.; Zeng, K.; Liu, J. Cyber–Physical Integrated Intrusion Detection Scheme in SCADA System of Process Manufacturing Industry. *IEEE Access* **2020**, *8*, 147471–147481. [CrossRef]
7. Sun, S.; Haque, K.A.; Huo, X.; Sahu, A.; Goulart, A.; Davis, K. Anomaly Detection in Cyber-Physical Systems Using Long-Short Term Memory Autoencoders: A Case Study with Man-in-the-Middle (MiTM) Attack. In *Proceedings of the 2025 IEEE Texas Power and Energy Conference (TPEC)*; IEEE: Piscataway, NJ, USA, 2025; pp. 1–6. [CrossRef]
8. Houkan, A.; Sahoo, A.K.; Gochhayat, S.P.; Sahoo, P.K. Artificial Intelligence Approach to Intrusion Detection in Industrial Control Systems with Real-World Dataset Generation and Model Evaluation. *Discov. Artif. Intell.* **2025**, *5*, 307. [CrossRef]
9. Walz, A.; Niemann, K.-H.; Göppert, J.; Fischer, K.; Merklin, S.; Ziegler, D.; Sikora, A. PROFINET Security: A Look on Selected Concepts for Secure Communication in the Automation Domain. In *Proceedings of the 2023 IEEE 21st International Conference on Industrial Informatics (INDIN)*; IEEE: Piscataway, NJ, USA, 2023; pp. 1–8. [CrossRef]
10. Niemann, K.-H.; Waldeck, B.; Esslinger, T. PROFINET–Future OT Security Requirements: What Do NIS2, CER, CRA and IEC 62443 Require? *atp Mag.* **2025**, *67*. [CrossRef]
11. Göppert, J.; Sikora, A. Evaluation of the Secure PROFINET Application Relation Establishment Performance. In *Proceedings of the 2024 IEEE 22nd International Conference on Industrial Informatics (INDIN)*; IEEE: Piscataway, NJ, USA, 2024; pp. 1–8. [CrossRef]
12. Mutlu, H.; Akyürek, N.; Korçak, Ö. PROFINET Controller on Cloud. In *Proceedings of the 24th International Symposium INFOTEH-JAHORINA*; IEEE: Piscataway, NJ, USA, 2025; pp. 1–6. [CrossRef]
13. Noorizadeh, M.; Shakerpour, M.; Meskin, N.; Unal, D.; Khorasani, K. A Cyber-Security Methodology for a Cyber-Physical Industrial Control System Testbed. *IEEE Access* **2021**, *9*, 16239–16253. [CrossRef]
14. Srivastava, A.; Zhao, J.; Moore, K.; Anagnostou, E. Development of Cyber Physical Security Simulation Testbed in RTDS Using Modbus Communication. In *Proceedings of the IEEE Power & Energy Society General Meeting (PESGM)*; IEEE: Piscataway, NJ, USA, 2024; pp. 1–6. [CrossRef]
15. Chen, B.; Pattanaik, N.; Goulart, A.; Butler-Purry, K.L.; Kundur, D. Implementing Attacks for Modbus/TCP Protocol in a Real-Time Cyber-Physical System Test Bed. In *Proceedings of the 2015 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT), Washington, DC, USA, 18–20 February*; IEEE: Piscataway, NJ, USA, 2015; pp. 1–6. [CrossRef]
16. Erceylan, G.; Akbarzadeh, A.; Gkioulos, V. Leveraging Digital Twins for Advanced Threat Modeling in Cyber Physical Systems Cybersecurity. *Int. J. Inf. Secur.* **2025**, *24*, 151. [CrossRef]
17. Lucchese, M.; Salerno, G.; Pugliese, A. A Digital Twin Based Approach for Detecting Cyber–Physical Attacks in ICS Using Knowledge Discovery. *Appl. Sci.* **2024**, *14*, 8665. [CrossRef]

18. Kosugi, A.; Teranishi, K.; Kogiso, K. Experimental Validation of the Attack Detection Capability of Encrypted Control Systems Using Man in the Middle Attacks. *IEEE Access* **2024**, *12*, 10535–10547. [CrossRef]
19. Yılmaz, E.N.; Sayan, H.H.; Üstünsoy, F.; Gönen, S.; Karacayılmaz, G. Cyber Security Analysis of DoS and MitM Attacks Against PLCs Used in Smart Grids. In *Proceedings of the 2019 IEEE International Conference on Smart Grids and Smart Cities (ICSGC), Istanbul, Turkey, 25–26 April 2019*; IEEE: Piscataway, NJ, USA, 2019; pp. 36–40. [CrossRef]
20. Raja, D.J.S.; Sriranjani, R.; Arulmozhi, P.; Hemavathi, N. Unified Random Forest and Hybrid Bat Optimization Based Man in the Middle Attack Detection in Advanced Metering Infrastructure. *IEEE Trans. Instrum. Meas.* **2024**, *73*, 2523812. [CrossRef]
21. Obonna, U.O.; Opara, F.K.; Mbaocha, C.C.; Obichere, J.-K.C.; Akwukwaegbu, I.O.; Amaefule, M.M.; Nwakanma, C.I. Detection of Man in the Middle (MitM) Cyber Attacks in Oil and Gas Process Control Networks Using Machine Learning Algorithms. *Future Internet* **2023**, *15*, 280. [CrossRef]
22. PCSimu: Industrial Automation Simulator (Software Version 3.0). 2023. Available online: https://cade-simu.com/pc-simu (accessed on 7 February 2026).
23. TIA Portal: STEP 7 Basic/Professional, STEP 7 Safety Basic/Advanced, WinCC Basic/Comfort/Advanced and WinCC Unified (Version 18.0). 2022. Available online: https://support.industry.siemens.com/cs/attachments/109807109/TIA_Portal_STEP7_Prof_Safety_WINCC_Prof_V18.iso (accessed on 7 February 2026).
24. Hilscher Gesellschaft für Systemautomation mbH. *PROFINET IO Device API—Protocol API.2. DocumentID: DOC190303API V5.8.0.1EN, Version 5.8.0.1*; Hilscher GmbH: Hattersheim am Main, Germany, 2025.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.