HENDRICS: A Hardware-in-the-Loop Testbed for Enhanced Intrusion Detection, Response and Recovery of Industrial Control Systems

Lalie Arnoud[1,2][0009−0003−3717−5250], Zoé Lagache[1,3][0009−0007−8238−988X], Pierre-Henri Thevenon[1][0000−0002−4390−5622], Aloïs Champenois[1][0009−0002−9622−3315], Victor Breux[1][0000−0001−9338−0185], Maxime Puys[4][0000−0001−6127−9816], Eric Gaussier[2][0000−0002−8858−3233], and Oum-El-Kheir Aktouf[3][0000−0002−0493−9096]

[1] Université Grenoble Alpes, CEA-Leti, F-38000 Grenoble, France
Firstname.Name@cea.fr
[2] Université Grenoble Alpes, CNRS, Grenoble INP, LIG, F-38000 Grenoble, France
Firstname.Name@univ-grenoble-alpes.fr
[3] Université Grenoble Alpes, Grenoble INP, LCIS, 26000 Valence, France
Firstname.Name@grenoble-inp.fr
[4] Université Clermont Auvergne, CNRS,
Clermont Auvergne INP, Mines Saint-Etienne, LIMOS, F-63000 Clermont-Ferrand, France
Firstname.Name@uca.fr

Abstract. Industrial Control Systems (ICS) automate processes across critical sectors like manufacturing, transport, and energy. The integration of Information Technology (IT) into historically isolated Operational Technology (OT) systems enhances efficiency but introduces vulnerabilities, particularly to cyberattacks such as those perpetrated by Advanced Persistent Threats. Intrusion detection systems, whether network-based (NIDS) or host-based (HIDS), are pivotal in safeguarding ICS, yet HIDS development faces challenges due to the closed-source nature of OT devices. To address these limitations, we present HENDRICS, an open-source testbed simulating a hydroelectric plant process. This platform embeds an open Programmable Logic Controller and an open Industrial IoT gateway in a Hardware-in-the-Loop (HitL) setup, enabling the research in the fields of intrusion detection, response, and recovery mechanisms, alongside being a realistic and affordable teaching platform. HENDRICS also provides 14 publicly available and documented attacks covering all Tactics and more than 67% of the Techniques from the MITRE ATT&CK for ICS matrix for the considered assets, encouraging reproducibility for advancing research and education. A comparative evaluation and future improvements are discussed.

Keywords: Industrial Control Systems (ICS), Testbed, Intrusion Detection Systems (IDS), MITRE ATT&CK

1 Introduction

Industrial Control Systems (ICS) are a set of components that enable an industrial process to be automated reliably and safely thanks to Operational Technologies (OT). ICSs are involved in a wide range of sectors, including chemical and manufacturing industries, transport and energy. The reference architecture often used for such systems follows the Purdue model [60], which proposes a hierarchical organization of these components. Sensors and actuators form the lowest level, as they directly operate the process. The second level contains Programmable Logic Controllers (PLCs), which control actuators on the basis of sensors values. The third level deals with process supervision, comprising interfaces for human operators. Higher levels are dedicated to task planning and overall site management.

More and more elements of Information Technology (IT) are being incorporated into ICSs to optimize plant performances. This enables more advanced analysis of field data, cross-referencing with other data sources, and incorporating data from the management system, *e.g.* production setpoints, into the decisions made by the ICS for greater efficiency. However, it brings together OT, historically isolated from the Internet and designed to remain in place for several decades, with IT, which faces other issues such as rapid evolution of technologies and vulnerability to cyberattacks. As a result, ICS, including those governing critical infrastructures, are increasingly vulnerable to cyberattacks perpetrated in particular by Advanced Persistent Threats (APTs) [46]. Intrusion detection, response and recovery mechanisms must therefore be studied to strengthen ICS resilience, since a compromised industrial infrastructure can lead to material, financial, ecological and human losses [21].

Most intrusion detection research currently relies on network traffic analysis to identify the presence of an intruder in a system [36]. These are known as Network-based Intrusion Detection Systems (NIDS). Besides, few studies focus on behavioral analysis of some devices in order to detect deviations from their normal behavior, or recognize known attack patterns, and thus be able to raise an alert if there are signs of intrusion [36]. Such systems are called Host-based Intrusion Detection Systems (HIDS). The main problem encountered in studying HIDS for industrial systems is that OT systems are usually closed-source. It means that we can very hardly instrument field devices like PLCs, Remote Terminal Units (RTUs) or data gateways in order to correctly identify their normal or compromised behavior from internal signals. The same constraint applies to attack response, system recovery, and other defense-in-depth mechanisms for these devices: it can be very difficult, if not impossible, to study such techniques without an appropriate platform.

To tackle this issue, we introduce our main contribution that is an affordable, easy-to-deploy, and **publicly available** testbed named HENDRICS, on which a PLC and an Industrial Internet of Things (I-IoT) gateway are integrated in a Hardware-in-the-Loop (HitL) environment simulating a hydroelectric plant industrial process, and can be fully instrumented to study defense-in-depth mechanisms. The proposed platform is described in Section 2. We then define a threat model in Section 3 and present our second main contribution, which is a set of **14 implemented or adapted attacks** to be executed on this platform, **publicly available** for the sake of reproducibility, and covering a large number of Tactics and Techniques of the MITRE ATT&CK

for ICS Matrix [13]. We compare our testbed against existing platforms in Section 4, according to various qualitative and quantitative criteria.Moreover, an analysis of the relevance of this platform for research and training purposes is made in Section 5. Finally, we summarize our contributions and discuss future enhancements in Section 6.

2 Proposed Testbed

HENDRICS focuses on representing the first three levels of the Purdue model [60]. An overview of our platform architecture is shown in Figure 1.

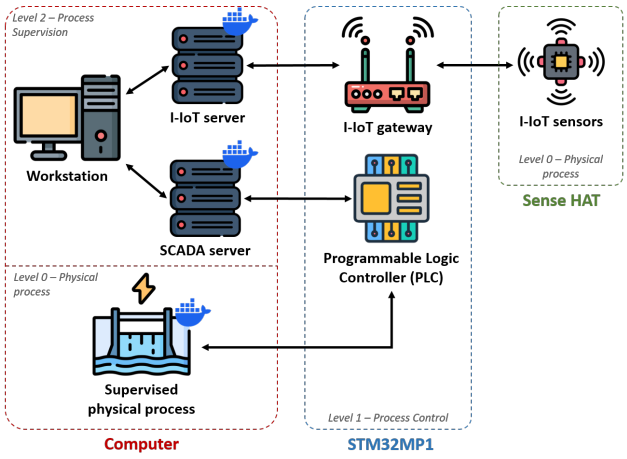


Fig. 1. High-level overview of the HENDRICS testbed

Sensors and actuators driving the hydroelectric power plant process are simulated on a computer. Also, two industrial devices abstractions, a PLC and an I-IoT gateway, can be both executed on a physical evaluation board to reproduce as faithfully as possible the internal functions of real industrial devices, in a Hardware-in-the-Loop manner. The PLC provides the logical control mechanisms that regulate the simulated industrial process, while the I-IoT gateway is responsible for feeding back information from the simple sensors of an add-on board to an I-IoT server. Finally, information related to the operational status of the industrial process and sensors are displayed in near-real time on web-based Human-Machine Interfaces (HMIs) which serve as Supervisory Control And Data Acquisition (SCADA) and I-IoT supervisors.

2.1 Hardware Overview

From a hardware perspective, the platform consists of at least one evaluation board, which is referenced as STMicroelectronics STM32MP157F-DK2 [8] and is based on STM32MP1 microprocessors, acting as an I-IoT gateway and/or a PLC. We

chose to use an STM32MP1 featuring a dual-core Cortex-A7 processor to closely match the technical characteristics of off-the-shelf industrial embedded systems. A Raspberry Pi Sense HAT board, connected to the device via an I^2C bus, provides environmental sensors and an LED matrix. Real-time data from sensors is sent by the I-IoT gateway to the I-IoT server and the temperature is displayed on the LED matrix. The simulated industrial process and the SCADA/I-IoT servers can both run on the same computer. STM32MP1 evaluation boards with their Sense HAT and the computer are connected via Ethernet to a single network managed by a switch. The total cost of the setup, based on average online prices rounded up to the nearest ten, is approximately 200 USD, assuming the user already has a computer for simulation and supervision. As such, HENDRICS is an affordable testbed comprising one or more physical industrial devices with a minimal number of hardware components.

2.2 Common Software Features

The STM32MP1 boards run a Linux 6.1 Super-Long-Term-Support (SLTS) kernel, maintained by the Civil Infrastructure Platform (CIP) project until 2033, ensuring alignment with current and future industrial embedded systems [32]. The kernel is PREEMPT-RT patched to provide soft real-time capabilities, enabling realistic device behavior. A lightweight Linux image is built using Buildroot, including BusyBox, a DropBear SSH server, and the necessary programs to operate as an I-IoT gateway or PLC, as further detailed in Sections 2.3 and 2.4.

2.3 I-IoT Gateway

The aim of this abstraction is to approximate the behavior of an Industrial IoT gateway, the role of which is to collect a set of environmental values from sensors, and send it to an I-IoT server for historization and monitoring. The I-IoT gateway and its environment is illustrated in Figure 2.

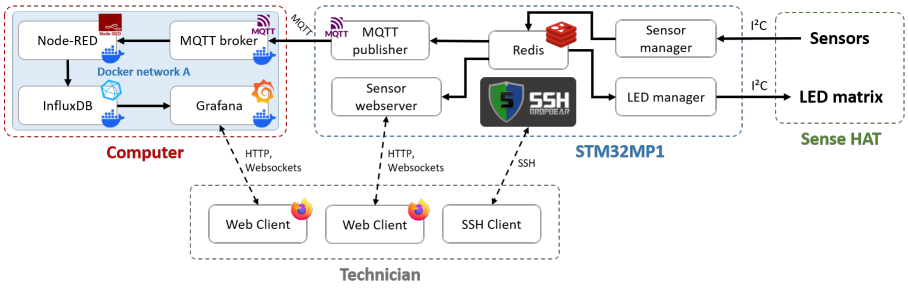


Fig. 2. I-IoT gateway and its environment

I-Iot Gateway Internal Operation. As mentioned in Section 2.1, HENDRICS relies on an STM32MP1 with a Sense HAT add-on board to act as an I-IoT gateway. The sensors on the add-on board (temperature, atmospheric pressure and humidity) are polled periodically via the I^2C bus, and their data is sent to Redis database. Redis functions as a publish-subscribe broker by acting as a publisher that immediately redistributes received data to its three subscribers:

- A webserver, written in Python using the Flask micro-framework, to which a technician can connect from a web browser to consult the measured sensor values as they become available, through WebSockets.
- A program that displays the temperature on the LED matrix of the Sense HAT according to measured values.
- A program that publishes this data to the MQTT broker of the I-IoT server for information gathering, also with a publish-and-subscribe logic.

I-IoT Server-side Information Gathering. Sensors data sent by the gateway to the server can be visualized as a function of time. Some tools are used to process and display such data:

- An MQTT broker, centralizing data published by the I-IoT gateway in the form of topics and sending it to subscribers of these channels. It enables communications through MQTT from the gateway to the server.
- Node-RED, subscribing to MQTT topics related to the Sense HAT sensor values and formatting it as database entries.
- InfluxDB, as a database storing data with its associated timestamps.
- Grafana, a tool constantly querying InfluxDB to display data as a web-based dashboard. A technician can then consult this interface using a web browser and view values of sensors on the Sense HAT evolving over time.

Each tool is encapsulated in a Docker image, *i.e.* an image of each application and its dependencies, so that they can run independently and isolated from each other at runtime on a separate Docker network. This facilitates the deployment and the reproducibility of the presented configuration, thanks to its construction and ease of use.

2.4 Programmable Logic Controller

This section describes components of HENDRICS reproducing the behavior of a Programmable Logic Controller (PLC). The PLC gathers data from simulated sensors to control simulated actuators, and sends the plant's state to the SCADA server for operator monitoring. Figure 3 illustrates the PLC and its environment.

Hydroelectric Plant Industrial Process. Unlike Sense HAT sensors, the industrial process in HENDRICS is entirely virtual, running on the same machine as the I-IoT and SCADA servers. The proposed physical process involves a continuously filling dam with two outputs: a safety valve (level-dependent) and a penstock with multiple valves channeling water to a turbine for electricity generation. Two pumps supply oil and water to cool the valves. The simulator, implemented in Python using

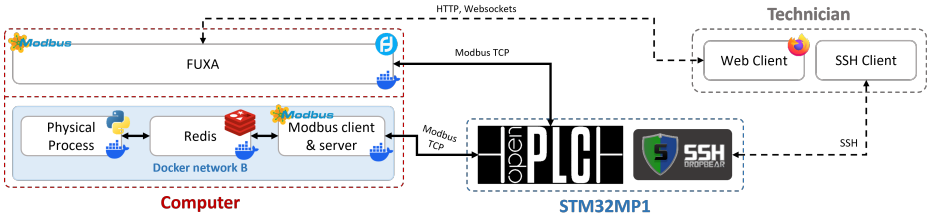


Fig. 3. PLC and its environment

the simpy [53] module, provides a process-based discrete-event simulation framework. Physical devices are modeled as independent and concurrently activated at regular intervals to compute outputs from inputs. This design allows software component reuse and allows interfacing with IT systems and industrial devices via standardized protocols. It is important to note that the industrial process simulator is interchangeable, the only prerequisites in current HENDRICS implementation being the ability to communicate with the simulated sensors and actuators via Modbus TCP, and knowledge of the PLC logic required for correct operation. As a result, we were able to use HENDRICS with the simplified version of the Tennessee-Eastman process simulator proposed by [30], known as GRFICS.

PLC Internal Operation. HENDRICS uses an STM32MP1 board to execute realistic PLC logic in *Structured Text* (ST), an IEC 61131-3 standard [48]. This program defines control rules for outputs (actuators) based on inputs (sensors). The PLC retrieves sensor data from the simulated plant, interprets the compiled ST program, updates outputs, sends I/O states to the SCADA server, and receives operator commands (*e.g.*, electricity production setpoint). These functions are handled by the OpenPLCv3 [15] runtime, which provides a compiler, execution engine, and a slave device configuration interface. Communication uses Modbus over TCP/IP. OpenPLC is a well-established choice in scientific literature for PLC logic interpretation [14,30,52,35], ensuring HENDRICS's relevance for realistic PLC behavior and ease of use. The evaluation board communicates with the hydroelectric plant simulator via Modbus TCP (sensors and actuators as servers) and with the SCADA server via another Modbus TCP connection.

SCADA Server. A **Supervisory Control And Data Acquisition (SCADA)** system visualizes industrial process states and controls technical installations. HENDRICS employs FUXA [4], an open-source tool, for web-based HMI monitoring. FUXA integrates a Modbus TCP client that queries the OpenPLC server for variables, animating the HMI accordingly.

3 Proposed ICS Attacks

In this section, we introduce a threat model and describe 14 distinct attacks that reflect real-world adversarial Tactics and Techniques from MITRE ATT&CK for ICS. The use of HENDRICS enables these attacks to be performed in a safe and representative way, bridging the gap between theoretical threat modeling and practical evaluation.

3.1 Threat Model

The core component of this testbed is a Linux-based embedded device comprising hardware (*e.g.* CPU, RAM, I/O), firmware, software applications, and networking capabilities which introduces potential attack vectors. To assess these threats, we analyze how individual components may be exploited by adversaries targeting embedded industrial systems such as PLC and industrial gateways. The threat landscape considered focuses primarily on software and network-level vulnerabilities that are commonly exploited in real-world industrial environments. Hardware-level threats, including side-channel attacks and fault injections, are excluded due to the assumption that the device is deployed in a controlled-access facility, where physical access requires authentication.

The most prevalent scenario involves an external adversary with access to the same network as the target device, often achieved through compromised IT systems, or misconfigured perimeter defenses that have allowed it to pivot within the internal network. In such cases, attackers can exploit unsecure-by-design industrial protocols like Modbus TCP which lack encryption and authentication, thereby exposing the system to data injection, replay, eavesdropping, or adversary-in-the-middle (AiTM) attacks. Accessible web dashboards or applications such as OpenPLC, Redis, and SSH, often configured with weak credentials, further increase the attack surface. These weaknesses can be leveraged for brute force or dictionary attacks on credentials, potentially allowing full control over the device or manipulation of critical process variables.

3.2 Developed Attacks

A set of 14 attacks targeting various elements of our PLC and I-IoT gateway was developed or adapted for the testbed. A brief description of each attack is provided below.

SSH dictionary attack. The attacker is able to obtain valid SSH credentials for the device, PLC or I-IoT gateway, by attempting common password combinations against a known username.

SSH AiTM. The attacker intercepts SSH communication between a technician and the device, PLC or I-IoT gateway. It begins with a reconnaissance step using Nmap [6]. Once the target is identified, Ettercap [3] spoofs ARP messages to redirect the connection to the attacker's machine. By initiating two SSH sessions and exchanging public keys with both technician and device, the attacker positions himself in the middle, enabling decryption of communications.

Valid SSH access gain via Redis. The attacker uses Hydra [59] to perform a dictionary attack on the Redis server to crack the password of a known user. Once authenticated, an RSA key pair is generated and the public key added to the server's authorized keys, granting the attacker SSH access to the I-IoT gateway.

BusyBox deletion via Redis. Hydra [59] is used against the Redis service to find the password of a known user. Once authenticated, a misconfiguration is exploited to change the server's working directory to `/bin`; the database dump will be placed in this directory. The dump is then renamed `busybox`, thus when the Redis service backs up its database, the BusyBox binary is overwritten, disabling all its commands and preventing the I-IoT gateway from functioning properly.

Redis DoS via Lua (CVE-2024-31228). Hydra [59] is used against the Redis service to find the password of a known Redis user. Once authenticated, a Lua script is executed on the Lua interpreter of the Redis client and performs a lot of operations on strings, which makes the program exceed its memory boundaries. The Redis process is then killed by the operating system.

Redis Lua injection via embedded webpage. The attacker exploits the I-IoT gateway's embedded web page, which technicians use to retrieve sensor values by entering a sensor name. The input is not sanitized, and the page generates a script executed on the Redis service. This vulnerability enables injection of queries into the internal Redis API, allowing manipulation of the Redis database.

Modbus AiTM. Since Modbus lacks encryption and authentication, an attacker on the same network can intercept, alter, and replay packets between OpenPLC and the SCADA, as well as between OpenPLC and the field I/O of the simulated hydroelectric plant. This causes false industrial process state values to be fed back to the SCADA system, misleading the technician about its real state, while it is impaired and causes, in our current implementation, the simulated dam to overflow.

OpenPLC Remote Code Execution (CVE-2021-31630). The attacker exploits the default OpenPLC credentials to log in. Once authenticated, the endpoint for uploading OpenPLC hardware driver is abused to load a reverse shell payload, which is interpreted and executed, granting the attacker access as the `openplc` user and stopping the PLC logic controlling the plant process.

Data exfiltration via ICMP. Hydra [59] is used against the Redis service to find the password of a known user. Once authenticated, the attacker uses a rogue Redis server that connect to the legitimate one on the I-IoT gateway and reconfigure it as its slave. The rogue server then sets the working directory of the legitimate server in a directory containing a periodically executed script, which will be overwritten at the first synchronization by a script that exfiltrates the `/etc/shadow` file via ICMP packets to the attacker's machine.

Data exfiltration via DNS. The attacker exploits default OpenPLC credentials to log in. Once logged in, the PLC program upload endpoint is abused to upload a Bash script that communicates with a previously configured fake DNS server [2], enabling exfiltration of the running PLC program. The attacker then uses the hardware driver upload endpoint to modify the driver so that it invokes the uploaded script without disrupting the PLC logic regulating the plant process.

Buffer overflow for privilege escalation. A vulnerable binary on the device, whose owner is `root` and which has the SUID bit, is exploited by making a buffer overflow and loading reverse shell shellcode to obtain a superuser shell.

PATH exploitation for privilege escalation. An existing binary which has the SUID bit calls another binary from the `PATH` environment variable. By hijacking the binary called and replacing it through the `PATH` variable by one that opens a shell, an attacker is able to get a shell with superuser privileges.

Boopkit [1]. Boopkit is an eBPF rootkit acting as a backdoor, spawning a reverse shell triggered by TCP ACK-RST packets. Communication is hidden from user space at kernel level, but installation requires root privileges.

Mirai [5]. The goal of Mirai is to infect embedded systems via a dictionary attack on well-know Telnet credentials and spread across networks to create a botnet used for launching Distributed Denial of Service (DDoS) attacks. It has three components: a Loader, controlled by the attacker, which connects to vulnerable devices and uploads the bot malware; a Command and Control (CNC) server, which tracks the bots and sends attack orders; and the bots, which scan their network to find other vulnerable devices and execute CNC-directed attacks.

3.3 Coverage of the MITRE ATT&CK for ICS Matrix

The MITRE ATT&CK for ICS Matrix [13] is a knowledge database built on the basis of reports of actual cyberattacks on industrial infrastructures. Positioned from the adversaries perspective, this matrix enables the various stages in the life cycle of a cyberattack to be retraced, from initial access to its concrete impact on the system. Attacks presented in this paper have been implemented in accordance with the Tactics, Techniques and Procedures (TTPs) of this matrix. This approach has been adopted to ensure that a wide range of Techniques is exploited and to provide a comprehensive representation of the mechanisms that can be used by an attacker to compromise PLCs and I-IoT gateways. Table 1 matches each of the 14 developed attacks with the Techniques used in its operation and the Techniques corresponding to its impact on our hydroelectric plant scenario. A quantitative measure of the diversity of the developed attacks can be computed, based on the number of Techniques implemented in our attack set out of the number of Techniques that can be implemented against the PLC (A0003) and Data Gateway (A0009) Assets, which fit our abstractions. Since the Assets proposed by MITRE ATT&CK do not define any Technique in the *Impact* Tactic, we do not consider this column for our evaluation. Consequently,

Table 1. Attacks implemented on HENDRICS with corresponding Techniques from MITRE ATT&CK for ICS v17 matrix

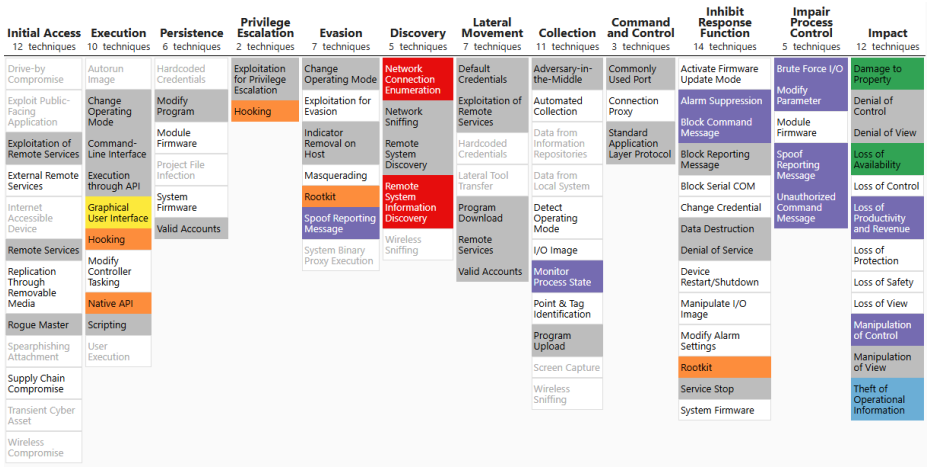
Attacks	Used Techniques from MITRE ATT&CK	Impacts from MITRE ATT&CK
SSH dictionary attack	T0859, T0886	
SSH AiTM	T0830, T0842, T0846, T0848, T0859, T0886	
Valid SSH access gain via Redis	T0807, T0859, T0886	
BusyBox deletion via Redis	T0807, T0809, T0814, T0881, T0886	
Redis DoS via Lua (CVE-2024-31228)	T0804, T0807, T0814, T0853, T0859, T0866, T0881, T0886	T0815
Redis Lua injection from embedded webpage	T0823, T0853, T0866	T0832
Modbus AiTM	T0801, T0803, T0804, T0806, T0830, T0836, T0842, T0848, T0855, T0856, T0878	T0813, T0815, T0828, T0831, T0832
OpenPLC RCE (CVE-2021-31630)	T0812, T0814, T0843, T0853, T0858, T0866, T0871, T0886, T0889	T0813, T0815, T0826, T0879
Data exfiltration via ICMP	T0845, T0848, T0853, T0859	
Data exfiltration via DNS	T0809, T0812, T0843, T0845, T0853, T0858, T0869, T0871, T0872, T0885, T0886, T0889	T0882
Buffer overflow for privilege escalation	T0807, T0853, T0890	
PATH exploitation for privilege escalation	T0807, T0853, T0890	
Boopkit	T0807, T0834, T0851, T0874	
Mirai	T0807, T0809, T0812, T0840, T0843, T0846, T0859, T0869, T0872, T0881, T0885, T0886, T0888	T0813, T0815

Underlined Techniques are those that only appear in the given attack. It can be seen that even if an attack does not cover more of the Techniques, it proposes a variant in the implementation of already used ones. This allows each of them to add value to the diversity of the set of attacks provided with the testbed.

Figure 4 shows that 43 Techniques are used out of 64 usable ones, representing 67.2% coverage. In addition to Table 1, this figure shows all TTPs used in the proposed attacks, highlighting their uniqueness. It also shows the Techniques that were not used and that could be the subject of future work.

3.4 Attack Scenarios based on Developed Attacks

From the individual attacks presented in Sections 3.2 and 3.3, more comprehensive attack scenarios can be constructed. Combining attacks increases the number of possible attacker behaviors, which is essential for assessing intrusion detection accuracy, response precision and recovery completeness. An example may be the deployment of the *boopkit* rootkit to provide a hidden backdoor that maintains access to a device. However, deploying such a program requires superuser privileges, which can be obtained by exploiting a *buffer overflow for privilege escalation*. Nevertheless, it requires access to the target, which can be obtained by exploiting a misconfiguration of the Redis server that leads to *valid SSH access gain via Redis*.



Shaded Techniques on a white background do not correspond to the MITRE ATT&CK PLC (A0003) and Data Gateway (A0009) Assets. They are therefore not relevant to our abstractions.
Techniques colored with a **gray background** are used by at least two implemented attacks.
Techniques colored with the following colors as background are used only by the specified attack:
purple: Modbus AiTM.
orange: boopkit.
red: Mirai.
green: OpenPLC RCE.
blue: Data exfiltration via DNS.
yellow: Redis Lua injection from embedded webpage.

Fig. 4. MITRE ATT&CK for ICS v17 Matrix coverage

4 Related work

In accordance with the qualitative criteria defined in the literature [20] and with our research challenges, we formulated four main requirements for comparison with related works. Table 2 analyzes existing state-of-the-art testbeds according to these requirements.

Fidelity. This criteria describes the ability of the testbed to replicate accurately real-world systems. Most existing research emphasizes network-level fidelity to analyze cyberattacks. In contrast, our focus is on embedded devices such as PLCs and I-IoT gateways, which are crucial for enhancing industrial systems resilience. To compare fidelity, we then consider OT components as outlined by the Purdue model [60]:

- **Level 0 – Physical Process, Field Devices**: (**R**)eal physical processes offer the highest fidelity, while simulated (**S**) processes simplify management and reduce risk, at the cost of realism.
- **Level 1 – Process Control, PLC**: Using commercial PLCs (*e.g.*, Schneider Electric, Siemens, ABB) ensures high fidelity but limits access to internal signals required for intrusion detection, attack response and device recovery. Virtual devices, although more accessible, have poor real-time responsiveness [14]. A satisfactory compromise could be the use of embedded boards running a Linux distribution and interpreting PLC logic. Hence PLCs can be categorized as:

  - **SW-S**: virtual devices with scripts that simulate PLCs behavior.
  - **SW-E**: virtual devices with software that executes real PLC logic.
  - **HW-E**: hardware devices with software that executes real PLC logic.
  - **HW-R**: commercial PLCs.
- **I-IoT features**: Industry 4.0 devices increasingly support IoT protocols such as MQTT or ZigBee. We evaluate whether these features are available (✓/ ✗) in each testbed.
- **Level 2 – Process Supervision, SCADA**: SCADA implementations are classified as (**C**)ommercial, (**O**)pen-source (*e.g.*, ScadaBR, FUXA [7,4]), or (**H**)omemade HMIs.

Replicability. Although setups with real processes and commercial PLCs offer realism, their high cost and lack of open-source availability limit their reproducibility for future research. Conversely, many testbeds use simulated processes and software-based PLCs, which are low-cost and often open-source. We assess replicability through estimated cost (Low [**$**]: 0–500 USD, Medium [**$$**]: 501–5,000 USD, High [**$$$**]: 5,001–50,000 USD), and source availability (GPLv3, MIT, Apache2.0, Mixed, ✗). The URL for accessing the sources is included in the references for each article featuring open-source software.

Scalability. This criteria describes the way in which a testbed allows the integration of future enhancements, such as extending the number of devices or adding communication protocols, without having to redesign the entire system.

Physical setups provide higher fidelity and enhance the credibility of research outcomes [44], they are limited in scalability, a challenge that virtualized testbeds can address. Then, scalability can be defined as follows:

- **Low (+)**: the testbed utilizes exclusively commercial and physical devices.
- **Medium (++)**: the testbed utilizes both virtual and physical components.
- **High (+++)**: the testbed is entirely virtual.

Attack diversity. The coverage of the MITRE ATT&CK for ICS Matrix describes the diversity of methods that can be used to compromise an ICS. To compute this coverage for each considered testbed, we count the number of Techniques used by attacks proposed by each work against the total number of Techniques available for the relevant Assets, often taken from the PLC Asset.

As can be seen from Table 2, the coverage of MITRE ATT&CK for ICS Matrix do not exceed 25% for literature testbeds, while ours reaches more than 67%. This difference can be attributed to the fact that these testbeds are primarily designed to validate normal system behavior and provide attack scenarios for proof-of-concept purposes, and thus are rarely intended to generate datasets that requires a higher degree of exhaustiveness in attack behaviors. López-Morales et al. [43] recently proposed the most comprehensive systematic survey on the security of PLCs. One major drawback highlighted in this article is the limited availability of reproducible research, with merely 16% of papers providing their research artifacts. They also

Table 2. Comparative analysis of ICS testbeds

		Fidelity				Replicability			
	Year	**Level 0**	**Level 1**	**I-IoT**	**Level 2**	**Licence**	**Cost**	**Scalability**	**ATT&CK coverage**
[58]	2015	S	SW-S	✗	H	✗	$	+++	16.1% (9/56)
[14]	2016	R+S	HW-R	✗	H	✗	$$	++	25% (14/56)
[44]	2016	R	HW-R	✗	C	✗	$$$	+	12.5% (7/56)
[30]	2018	S	SW-E	✗	H	GPLv3	$	+++	0%
[56]	2018	R	HW-R	✗	✗	✗	$$	+	16.1% (9/56)
[37]	2019	R+S	HW-R	✗	C	✗	$$	+	0%
[52]	2019	R	HW-E	✗	O	Mixed	$$	+	23.2% (13/56)
[34]	2019	R+S	HW-R	✓	✗	✗	$$$	+	12.5% (8/64)
[12]	2020	R	HW-E	✓	H	MIT	$	++	20.3% (13/64)
[47]	2021	S	HW-R	✗	C	✗	$$$	+	0%
[33]	2021	R	HW-E	✗	H	✗	$$	++	12.5% (7/56)
[49]	2021	S	HW-R	✗	C	✗	$$$	+	19.6% (11/56)
[27]	2023	S	SW-S	✗	H	MIT	$	+++	19.6% (11/56)
[28]	2023	R	SW-E	✗	H	✗	$	+++	16.1% (9/56)
[35]	2024	S	SW-E	✗	O	Apache2.0	$	+++	23.2% (13/56)
Our work	**2025**	**S**	**HW-E**	✓	**O**	**GPLv3**	**$**	++	**67.2% (43/64)**

Level 0 - Physical process: (**R**)eal, (**S**)imulation.
Level 1 - PLC: **SW-S**: virtual device with common scripts; **SW-E**: virtual device with PLC software; **HW-E**: embedded device with PLC software; **HW-R**: commercial PLC.
I-IoT features: Available (✓) or not (✗).
Level 2 - SCADA: (**C**)ommercial SCADA software, (**O**)pen source software, or (**H**)omemade HMI.
Licence: Open-source licence of the source code; (✗) if sources are not available.
Cost: Low [**$**]: 0-500 USD, Medium [**$$**]: 501-5,000 USD, or High [**$$$**]: 5,001-50,000 USD.
Scalability: Low [+]: only physical components; Medium [++]: both virtual and physical components; or High [+++]: only virtual components.
ATT&CK coverage: Number of Techniques from the MITRE ATT&CK for ICS Matrix for the Assets available on each considered testbed, and the percentage coverage of the matrix. It is important to note that these values are derived from our comprehension of attacks presented in each paper. This estimation only takes into account attacks for which there is an explicit evidence that an implementation has been made.

demonstrates that a significant number of ATT&CK Tactics have not been explored in state-of-the-art researches, such as *Command and Control*, *Lateral Movement*, *Evasion* and *Initial Access*. By making the source code available, attacks easily reproducible, and proposing the best coverage of Tactics (100%) and Techniques (67.2%) from the

MITRE ATT&CK for ICS Matrix considering the PLC and Data Gateway Assets, our testbed fills a clearly identified research gap. Hence, HENDRICS is the most complete platform in the literature to be designed for the development of host-based intrusion detection, attack responses and recovery mechanisms for industrial control systems.

5 Usages for ICS Resilience

HENDRICS can be useful for ICS hardening research, but it can also be used in multiple contexts such as a cybersecurity assessment training and awareness platform, offering valuable benefits due to its open-source and low-cost nature.

5.1 Research Topics

Intrusion Detection. The platform presented in this paper can be a valuable asset for building and evaluating intrusion detection mechanisms in industrial environments. Indeed, the role of Intrusion Detection Systems (IDS) is to monitor events that occur on a computer or a network and analyze them for signs of malicious actions [39]. They usually require a training phase that would allow them to distinguish between a normal behavior and an abnormal one, and an evaluation phase to validate the efficiency of their detection. A set of data extracted from realistic behavior under normal conditions and during attacks is therefore essential for IDS development, as both the training and evaluation phases need data to properly assess the theoretical performance of such detection methods close to real-world situations.

To the best of our knowledge, state-of-the-art host-based IDS are mostly trained and evaluated on aging datasets such as UNM [31], DARPA98 [38] and ADFA-LD [22]. However, these datasets have issues that the proposed platform aims to tackle. First, these datasets were collected on hosts whose kernel support was discontinued in 2003, 2005 and 2011 respectively [55,40]. The Linux kernel used in our platform is an SLTS 6.1 kernel whose end-of-life is scheduled by the Linux Foundation for December 2026 [41] and extended by the CIP project to 2033 [32]. We believe this makes our platform an effective and up-to-date representation of recent industrial systems. Secondly, these datasets have not been extracted from platforms representing the operational part of industrial systems. Consequently, the behaviors recorded in these datasets are not representative of the behaviors we may find in real industrial systems. Moreover, the range of attacks in these datasets is rather limited and does not cover the specific attacks possible on an ICS. Our work fills this gap by proposing an instrumentable ICS testbed for such intrusion detection experiments as well as a set of attacks whose diversity and closeness to real attackers techniques are measured by the coverage of the MITRE ATT&CK for ICS Matrix. Thirdly, as our testbed is physical and not virtual, it enables gathering realistic behavioral information from the host under study, like hardware performance counters [25] and system calls [42], which would make it possible to properly compare the performance of these different intrusion detection methods on the same environment.

HENDRICS can thus be a suitable platform to generate benchmark datasets for host-based intrusion detection on ICS, thanks to both its extensive normal and under attack behaviors, and its up-to-date software.

System Response and Recovery. The proposed testbed is a key step toward enhancing ICS resilience by offering a dedicated platform to test and validate attack response and recovery strategies, core functions of the NIST Cybersecurity Framework [45]. It enables researchers and manufacturers to develop techniques that respond or recover from attacks like those in Table 1, such as checkpointing the state of the system for subsequent rollback [18], malware isolation [57,26], or hot patching [50]. HENDRICS allows safe, real-time testing of these techniques. Attack scenarios as illustrated in Section 3.4, offer a wide range of threats to assess the efficiency of a solution orchestrating several recovery and response techniques against complex attacks. By mapping our testbed attacks to MITRE ATT&CK, we provide a standardized method to describe adversarial behaviors. This allows for the separation of response and recovery steps from detection, enabling strategies tailored to ATT&CK-defined techniques, while many existing solutions are tailored for their custom systems [19,61,11], limiting their compatibility with ICS scenarios. Tools like Wazuh illustrate how integrating TTPs from MITRE ATT&CK can facilitate standardized threat information exchange [9]. This demonstrates the potential for designing response and recovery processes independently of specific IDS solutions through the use of such frameworks. The platform also offers strong coverage of the ICS ATT&CK Matrix (Table 2), supporting diverse response and recovery scenarios under realistic threat conditions.

5.2 Training Platform for Cybersecurity Assessment

Education. Cybersecurity courses are now an important part of computer science curricula. In particular, ICS resilience is a major concern for government agencies and giving such knowledge to students is seen as a valuable experience [23,29,17,24]. Teaching in this field is particularly challenging as ICSs are usually made of costly and proprietary equipment, requiring to purchase licences every year. As a low-cost open-source platform, HENDRICS is well-suited for educational purposes and can be used to demonstrate cybersecurity incidents during lectures or as a lab session platform. Indeed, HENDRICS platform could ease several educational exercises, such as: discovering the structure of ICSs, manipulating ICS protocols (Modbus protocol dissection, frame injection, etc.), understanding the impact of attacks on the physical process, learning to detect cyberattacks with intrusion detection solutions such as NIDS, HIDS, and more.

Industry. In industry, cybersecurity awareness is a critical aspect in the prevention of attacks against industrial facilities [54,16,49]. HENDRICS can also be a particularly useful platform for ICS equipment vendors and ICS facility owners. HENDRICS relies on OpenPLC, a well-known open-source PLC implementation, allowing engineers to train, develop and test detection tools and response strategies against live complex attacks. Moreover, HENDRICS offers an environment comprising real-world industrial protocols[5] which could be applied to several industrial use cases. It also provides an industrial process simulation with more than 40 controlled variables describing a safety critical use case, a hydroelectric plant, enabling for instance red team/blue team exercises to be conducted by industrial engineers.

[5] HENDRICS currently only implements Modbus, but OpenPLC supports Modbus, DNP3, OPC-UA with [35] and IEC61850 from [51], allowing for future enhancements.

6 Conclusion

The number of cyberattacks on industrial control systems is increasing, particularly due to the convergence of operational and information technologies. The development of defense-in-depth mechanisms is therefore a priority, and requires suitable platforms for the development of intrusion detection, attack response and recovery systems.

In this paper, we present HENDRICS, an open-source, low-cost and scalable testbed integrating instrumentable PLCs and I-IoT gateways in a proposed simulated hydroelectric plant environment whose use case can be changed. It stands out from the already existing ICS platforms by its accessibility, reproducibility and fidelity to a real industrial system, making it an interesting compromise between the cost of such a platform and the credibility of the research and teaching that will be carried out on it. Moreover, the platform is flexible and can be easily adapted to other industrial use cases through its physical process simulation. The second main contribution is a large set of reproducible attacks for this testbed, covering a wide range of scenarios. These attack scenarios represent more than 67% of the Techniques listed in the MITRE ATT&CK for ICS Matrix for the PLC and Data Gateway Assets, which is significantly higher than the works available in the literature. By replicating real-world conditions, this comprehensive approach enables a realistic representation of the threats faced by modern industrial systems, providing a robust foundation for researchers and engineers to develop and evaluate security solutions under controlled and repeatable attack scenarios.

In future work, we plan to extend the testbed by integrating additional industrial and wireless communication protocols to support the development of realistic attack scenarios aligned with the MITRE ATT&CK framework. We will also implement tools for extracting internal signals, such as system calls and hardware performance counters, to enable the generation of high-quality datasets for intrusion detection.

Open science. We encourage the use of HENDRICS by making the source code of the entire testbed, namely, the source code for the device (both I-IoT gateway and PLC assets), for I-IoT and SCADA servers, for the proposed hydroelectric process simulator, and for the 14 presented attacks, publicly available at [10] under the GPLv3 licence. Technical documentation describing the steps involved in compiling, deploying and attacking the testbed are also provided. However, attacks presented in this paper are **intended to be used only on this testbed** for educational and research purposes as they can be dangerous for connected devices. We therefore strongly advise anyone wishing to reproduce our work to do so as described in this paper, namely on a **closed, independent and isolated network**.

Acknowledgments. We would like to express our gratitude to Antoine Cavassila-Beuf, Antoine Chapel, Hugo Hoarau and Emmanuel Brun for their important contributions during the design and implementation of this testbed. This work was supported by the French National Research Agency in the framework of the "Investissements d'avenir" program (IRT Nanoelec, ANR-10-AIRT-05). This work has been partially supported by the French National Research Agency under the France 2030 labels (Superviz ANR-22-PECY-0008, MIAI Cluster ANR-23-IACL-0006). The views reflected herein do not necessarily reflect the opinion of the French government.

References

1. Boopkit. https://github.com/krisnova/boopkit, accessed: 2025-09-04
2. dnsteal. https://github.com/m57/dnsteal, accessed: 2025-09-04
3. Ettercap. https://www.ettercap-project.org, accessed: 2025-09-04
4. FUXA – Web-based Process Visualization (SCADA/HMI/Dashboard) software. https://github.com/frangoteam/FUXA, accessed: 2025-09-04
5. Mirai Source Code. https://github.com/jgamblin/Mirai-Source-Code, accessed: 2025-09-04
6. Nmap: the Network Mapper). https://nmap.org/, accessed: 2025-09-04
7. ScadaBR - Open source SCADA. https://github.com/ScadaBR, accessed: 2025-09-04
8. Stm32mp157f-dk2 – discovery kit with stm32mp157f mpu. https://www.st.com/en/evaluation-tools/stm32mp157f-dk2.html, accessed: 2025-09-04
9. Wazuh – mitre att&ck framework. https://documentation.wazuh.com/current/user-manual/ruleset/mitre.html, accessed: 2025-09-04
10. Hendrics: A hardware-in-the-loop testbed for enhanced intrusion detection, response and recovery of industrial control systems (source code). https://github.com/CEA-Leti/HENDRICS/tree/release-anubis (2025), accessed: 2025-01-22
11. Abdulrazak, B., Codjo, J.A., Paul, S.: Self-healing Approach for IoT Architecture: AMI Platform, Lecture Notes in Computer Science, vol. 13287, p. 3–17. Springer International Publishing, Cham (2022). https://doi.org/10.1007/978-3-031-09593-1_1
12. Al-Hawawreh, M., Sitnikova, E.: Developing a Security Testbed for Industrial Internet of Things. IEEE Internet of Things Journal **8**, 5558–5573 (2020). https://doi.org/0.48550/arXiv.2011.12546, source-code: https://github.com/Alhawawreh/Brown-IIoTbed
13. Alexander, O., Belisle, M., Steele, J.: MITRE ATT&CK for industrial control systems: Design and philosophy. The MITRE Corporation: Bedford, MA, USA **29** (2020)
14. Alves, T., Das, R., Morris, T.: Virtualization of Industrial Control System Testbeds for Cybersecurity. In: Proceedings of the 2nd Annual Industrial Control System Security Workshop. pp. 10–14. ICSS '16, Association for Computing Machinery, New York, NY, USA (2016). https://doi.org/10.1145/3018981.3018988
15. Alves, T.R., Buratto, M., de Souza, F.M., Rodrigues, T.V.: Openplc: An open source alternative to automation. In: IEEE Global Humanitarian Technology Conference (GHTC 2014). pp. 585–589 (2014). https://doi.org/10.1109/GHTC.2014.6970342
16. Angafor, G.N., Yevseyeva, I., He, Y.: Bridging the cyber security skills gap: Using tabletop exercises to solve the cssg crisis. In: Ma, M., Fletcher, B., Göbel, S., Baalsrud Hauge, J., Marsh, T. (eds.) Proc. of the 15th Joint International Conference on Serious Games, Staffordshire University, Stoke-on-Trent, UK, LNCS. Lecture Notes in Computer Science, vol. 12434, pp. 117–131. Springer-Verlag (Nov 2020)
17. de la Sécurité des Systèmes d'Information (ANSSI), A.N.: Organising a cyber crisis management exercise. https://cyber.gouv.fr/sites/default/files/2021/09/anssi-guide-organising_a_cyber_crisis_management_exercise-v1.0.pdf (2021), accessed: 2025-09-04
18. Bashiri, M., Miremadi, S.G., Fazeli, M.: A Checkpointing Technique for Rollback Error Recovery in Embedded Systems. In: 2006 International Conference on Microelectronics. pp. 174–177. IEEE (2006). https://doi.org/10.1109/ICM.2006.373295
19. Chevalier, R., Plaquin, D., Dalton, C., Hiet, G.: Intrusion Survivability for Commodity Operating Systems. Ph.D. thesis, Université Paris-Saclay (2020)
20. Conti, M., Donadel, D., Turrin, F.: A Survey on Industrial Control System Testbeds and Datasets for Security Research. IEEE Communications Surveys & Tutorials **23**, 2248–2294 (2021), url: https://api.semanticscholar.org/CorpusID:231861910

21. Corallo, A., Lazoi, M., Lezzi, M.: Cybersecurity in the context of industry 4.0: A structured classification of critical assets and business impacts. Computers in Industry **114**, 103165 (2020). https://doi.org/https://doi.org/10.1016/j.compind.2019.103165
22. Creech, G., Hu, J.: A Semantic Approach to Host-Based Intrusion Detection Systems Using Contiguousand Discontiguous System Call Patterns. IEEE Transactions on Computers **63**(4), 807–819 (Apr 2014). https://doi.org/10.1109/TC.2013.13
23. Cybersecurity, (CISA), I.S.A.: Ics cybersecurity training available through cisa. https://www.cisa.gov/resources-tools/programs/ics-training-available-through-cisa (2025), accessed: 2025-09-04
24. for Cybersecurity (ENISA), E.U.A.: Cybereducation platform. https://www.enisa.europa.eu/topics/education-and-career-path/cybereducation-platform (2025), accessed: 2025-09-04
25. Das, S., Werner, J., Antonakakis, M., Polychronakis, M., Monrose, F.: SoK: The Challenges, Pitfalls, and Perils of Using Hardware Performance Counters for Security. In: 2019 IEEE Symposium on Security and Privacy (SP). pp. 20–38. IEEE, San Francisco, CA, USA (May 2019). https://doi.org/10.1109/SP.2019.00021
26. DaSilva, P.R., Fortier, P.J.: Hardware Based Detection, Recovery, and Tamper Evident Concept to Protect from Control Flow Violations in Embedded Processing. In: 2019 IEEE International Symposium on Technologies for Homeland Security (HST). pp. 1–6 (2019). https://doi.org/10.1109/HST47167.2019.9032955
27. Dehlaghi-Ghadim, A., Balador, A., Moghadam, M.H., Hansson, H., Conti, M.: ICSSIM — A framework for building industrial control systems security testbeds. Computers in Industry **148**, 103906 (2023). https://doi.org/10.1016/j.compind.2023.103906, source-code: https://github.com/AlirezaDehlaghi/ICSSIM
28. Erkek, I., Irmak, E.: Evaluation of SCADA Test Beds and Design of a New Software-Based Test Bed. In: 2023 11th International Conference on Smart Grid (icSmartGrid). pp. 1–8 (2023). https://doi.org/10.1109/icSmartGrid58556.2023.10170935
29. (FEMA), F.E.M.A.: Critical infrastructure security and resilience training. https://training.fema.gov/is/cisr.aspx (2025), accessed: 2025-09-04
30. Formby, D., Rad, M., Beyah, R.: Lowering the Barriers to Industrial Control System Security with GRFICS. In: 2018 USENIX Workshop on Advances in Security Education (ASE 18). USENIX Association, Baltimore, MD (Aug 2018), source-code: https://github.com/djformby/GRFICS
31. Forrest, S., Hofmeyr, S., Somayaji, A., Longstaff, T.: A sense of self for Unix processes. In: Proceedings 1996 IEEE Symposium on Security and Privacy. pp. 120–128 (1996). https://doi.org/10.1109/SECPRI.1996.502675
32. Foundation, T.L.: Civil infrastructure platform. https://wiki.linuxfoundation.org/civilinfrastructureplatform/start#kernel_maintainership, accessed: 2025-09-04
33. Fujita, S., Hata, K., Mochizuki, A., Sawada, K., Shin, S., Hosokawa, S.: OpenPLC based control system testbed for PLC whitelisting system. Artificial Life and Robotics **26**(1), 149–154 (Feb 2021). https://doi.org/10.1007/s10015-020-00635-1
34. Gardiner, J., Craggs, B., Green, B., Rashid, A.: Oops I Did it Again: Further Adventures in the Land of ICS Security Testbeds. In: CPS-SPC'19: Proceedings of the ACM Workshop on Cyber-Physical Systems Security & Privacy. pp. 75–86 (Nov 2019). https://doi.org/10.1145/3338499.3357355
35. Giehl, A., Heinl, M.P., Embacher, V.: EmuFlex: A Flexible OT Testbed for Security Experiments with OPC UA. In: Proceedings of the 19th International Conference on Availability, Reliability and Security. ARES '24, Association for Computing Machinery, New York, NY, USA (2024). https://doi.org/10.1145/3664476.3670931, source-code: https://github.com/Fraunhofer-AISEC/ot-security-testbed

36. Khraisat, A., Gondal, I., Vamplew, P., Kamruzzaman, J.: Survey of intrusion detection systems: Techniques, datasets and challenges. Cybersecurity **2**(1), 20 (Dec 2019). https://doi.org/10.1186/s42400-019-0038-7
37. Krishnan, S., Wei, M.: SCADA Testbed for Vulnerability Assessments, Penetration Testing and Incident Forensics. In: 2019 7th International Symposium on Digital Forensics and Security (ISDFS). pp. 1–6 (2019). https://doi.org/10.1109/ISDFS.2019.8757543
38. Laboratory, M.L.: 1998 darpa intrusion detection evaluation dataset. https://www.ll.mit.edu/r-d/datasets/1998-darpa-intrusion-detection-evaluation-dataset, accessed: 2025-09-04
39. Liao, H.J., Richard Lin, C.H., Lin, Y.C., Tung, K.Y.: Intrusion detection system: A comprehensive review. Journal of Network and Computer Applications **36**(1), 16–24 (Jan 2013). https://doi.org/10.1016/j.jnca.2012.09.004
40. (via the Linux Kernel Mailing List Archive), G.K.: Linux 2.6.38.8. https://lkml.iu.edu/hypermail/linux/kernel/1106.0/01226.html, accessed: 2025-09-04
41. Linux Kernel Organization, I.: The linux kernel archives - releases. https://www.kernel.org/category/releases.html, accessed: 2025-09-04
42. Liu, M., Xue, Z., Xu, X., Zhong, C., Chen, J.: Host-Based Intrusion Detection System with System Calls: Review and Future Trends. ACM Computing Surveys **51**(5), 1–36 (Sep 2019). https://doi.org/10.1145/3214304
43. López-Morales, E., Planta, U., Rubio-Medrano, C., Abbasi, A., Cardenas, A.A.: SoK: Security of programmable logic controllers. In: 33rd USENIX Security Symposium (USENIX Security 24). pp. 7103–7122. USENIX Association, Philadelphia, PA (Aug 2024)
44. Mathur, A., Tippenhauer, N.O.: SWaT: a water treatment testbed for research and training on ICS security. pp. 31–36 (Apr 2016). https://doi.org/10.1109/CySWater.2016.7469060
45. National Institute of Standards and Technology: NIST Cybersecurity Framework. url: https://www.nist.gov/cyberframework (2023), accessed: 2025-09-04
46. Nicholson, A., Webber, S., Dyer, S., Patel, T., Janicke, H.: SCADA security in the light of Cyber-Warfare. Computers & Security **31**(4), 418–436 (Jun 2012). https://doi.org/10.1016/j.cose.2012.02.009
47. Ozçelik, I., Iskefiyeli, M., Balta, M., Akpinar, K.O., Toker, F.S.: CENTER Water: A Secure Testbed Infrastructure Proposal For Waste and Potable Water Management. In: 2021 9th International Symposium on Digital Forensics and Security (ISDFS). pp. 1–7 (2021). https://doi.org/10.1109/ISDFS52919.2021.9486364
48. PLCOpen: IEC 61131-3: a standard programming resource. https://plcopen.org/sites/default/files/downloads/intro_iec_oct2016.pdf, accessed: 2025-09-04
49. Puys, M., Thevenon, P.H., Mocanu, S.: Hardware-In-The-Loop Labs for SCADA Cybersecurity Awareness and Training. pp. 1–10 (Aug 2021). https://doi.org/10.1145/3465481.3469185
50. Rajput, P.H.N., Doumanidis, C., Maniatakos, M.: Icspatch: Automated vulnerability localization and non-intrusive hotpatching in industrial control systems using data dependence graphs (Dec 2022), arXiv:2212.04229
51. Roomi, M.M., Ong, W.S., Hussain, S.M.S., Mashima, D.: Iec 61850 compatible openplc for cyber attack case studies on smart substation systems. IEEE Access **10**, 9164–9173 (2022). https://doi.org/10.1109/ACCESS.2022.3144027
52. Sauer, F., Niedermaier, M., Kießling, S., Merli, D.: LICSTER -A Low-cost ICS Security Testbed for Education and Research (Sep 2019). https://doi.org/10.14236/ewic/icscsr19.1, source-code: https://github.com/thainnos/LICSTER
53. Scherfke, S., Lünsdorf, O.: Simpy, discrete event simulation for python. https://simpy.readthedocs.io/, accessed: 2025-09-04

54. Sitnikova, E., Foo, E., Vaughn, R.B.: The power of hands-on exercises in SCADA cyber security education. In: Dodge, R.C., Futcher, L. (eds.) Proc. of the 8th IFIP WG 11.8 World Conference on Information Security Education (WISE 8), Auckland, New Zealand. IFIP Advances in Information and Communication Technology, vol. 406, pp. 83–94. Springer-Verlag (Jul 2013)
55. Sun Microsystems, I.: End of service life status for solaris operating system. https://web.archive.org/web/20101108171649/www.sun.com:80/service/eosl/eosl_solaris.html, accessed: 2025-09-04
56. Teixeira, M., Salman, T., Zolanvari, M., Jain, R.: SCADA System Testbed for Cybersecurity Research Using Machine Learning Approach. Future Internet **10** (Aug 2018). https://doi.org/10.3390/fi10080076
57. Thomas, Z., Abdelwahed, S.: Active malware countermeasure approach for mission critical systems. In: 2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech). p. 632–638. IEEE, Orlando, FL (Nov 2017). https://doi.org/10.1109/DASC-PICom-DataCom-CyberSciTec.2017.111
58. Thornton, J.Z.: A virtualized SCADA laboratory for research and teaching (2015), url: https://api.semanticscholar.org/CorpusID:115014434
59. Van, H.: Hydra. https://github.com/vanhauser-thc/thc-hydra, accessed: 2025-09-04
60. Williams, T.J.: A Reference Model for Computer Integrated Manufacturing from the Viewpoint of Industrial Automation. IFAC Proceedings Volumes **23**(8, Part 5), 281–291 (1990). https://doi.org/10.1016/S1474-6670(17)51748-6
61. Yagemann, C., Pruett, M., Chung, S.P., Bittick, K., Saltaformaggio, B., Lee, W.: Arcus: Symbolic root cause analysis of exploits in production systems. In: 30th USENIX security symposium (USENIX security 21). p. 1989–2006. USENIX Association (Aug 2021)