Hardware-In-The-Loop Labs for SCADA Cybersecurity Awareness and Training

Maxime Puys
Pierre-Henri Thevenon
Univ. Grenoble Alpes, CEA, LETI, DSYS, LSOSP, F-38000
Grenoble, France
Firstname.Name@cea.fr

Stéphane Mocanu
Laboratoire d'Informatique de Grenoble
Univ. Grenoble Alpes, CNRS, Inria, Grenoble-INP
Grenoble, France
Stephane.Mocanu@imag.fr

ABSTRACT

In this paper, we present a SCADA cybersecurity awareness and training program based on a Hands-On training using two twin cyber-ranges named WonderICS and G-ICS. These labs are built using a Hardware-In-the-Loop simulation system of the physical process developed by the two partners. The cyber-ranges allow replication of realistic Advanced Persistent Threat (APT) attacks and demonstration of known vulnerabilities, as they rely on real industrial control devices and softwares. In this work, we present both the demonstration scenarios used for awareness on WonderICS and the training programs developed for graduate students on G-ICS.

KEYWORDS

Industrial Control Systems Security, SCADA, Critical Infrastructure

ACM Reference Format:
Maxime Puys, Pierre-Henri Thevenon, and Stéphane Mocanu. 2021. Hardware-In-The-Loop Labs for SCADA Cybersecurity Awareness and Training . In *The 16th International Conference on Availability, Reliability and Security (ARES 2021), August 17–20, 2021, Vienna, Austria.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3465481.3469185

1 INTRODUCTION

Industrial Control Systems (ICS) are network interconnected hardware and software components designed to control and operate physical plants or industrial facilities. Historically deployed on dedicated and isolate proprietary networks, they were considered protected from networks threats. This "protection by isolation and obscurity" myth ended in 2010 when Stuxnet malware attacked Iranian plants in Natanz [10]. From this point, cyber-incidents concerning ICS were regularly discovered and the number of attacks against industrial facilities is continuously growing. The deployment of Internet technologies in ICS exposed them to remote threats and interconnection with general IT make them vulnerable to general purpose malware such as ransomwares. Nowadays, cybersecurity of ICS can't be ignored anymore and deployment of an Information Security System (ISS) and, consequently, security controls, is mandatory for some critical industrial activities (for instance the operators of essential services as defined by the European Directive on Security of Network and Information Systems [6]). In particular, one key security control is relevant for the present work: Awareness and Training. Indeed, a large proportion of cyber-incidents will exploit the lack of user awareness of computer and Internet risks. A statistical study published in 2019 by the German Federal Office for Information Security (BSI) showed that among the top ten threats used by attackers to penetrate ICS (see Table 1 from [2]), at least six can be addressed with an adequate user awareness and training in complement to technical controls (Infiltration via Removable Media, Infection via Internet and Intranet, Human Error and Sabotage, Social Engineering and Phishing, Intrusion via Remote Access, Compromising of Smartphones in the Production Environment).

Infiltration via Removable Media and External Hardware
Malware Infection via Internet and Intranet
Human Error and Sabotage
Compromising of Extranet and Cloud Components
Social Engineering and Phishing
(D)Dos Attacks
Control Components Connected to the Internet
Intrusion via Remote Access
Technical Malfunctions and Force Majeure
Compromising of Smartphones in the Production Environment

Table 1: Top 10 threats against ICS in 2019 [2].

Most of the industrial systems cybersecurity deployment guides recommend awareness and training as a first control to be deployed. See for instance the on-line training resources of ICS-CERT[1] or the French agency ANSSI ICS training guide[2]. In particular, hands-on training and attack scenario demonstrators are proven to be the most effective tools [1, 14]. The main challenge in industrial systems cybersecurity training is to create a realistic environment.

This work is supported by the French National Research Agency in the framework of the "Programme d'Investissement d'Avenir IRT Nanoelec" (ANR-10-AIRT-05) and "Investissements d'avenir IDEX UGA" (ANR-15-IDEX-02).

ARES 2021, August 17–20, 2021, Vienna, Austria
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-9051-4/21/08…$15.00
https://doi.org/10.1145/3465481.3469185

[1] https://us-cert.cisa.gov/ics/Training-Available-Through-ICS-CERT
[2] https://www.ssi.gouv.fr/entreprise/guide/guide-pour-une-formation-sur-la-cybersecurite-des-systemes-industriels/

Obviously, it is not possible to play attacks on a real facility and reproducing a physical process of realistic size in an university lab is very costly and challenging or can simply be impossible in case of a dangerous process (e.g., chemical factory). Several cyber-range initiatives were dedicated to research and training. Probably the most famous one is the Idaho National Laboratories (INL) which reproduce a real electrical substation dedicated to smart-grid cyber-security assessment and training [9]. Other initiatives concerned commercial virtual cyber-ranges like CRIAB[3] proposed by Boeing or the European Airbus[4].

Fully virtual cyber-ranges are flexible and clearly less difficult to maintain than a lab reproduction of a real industrial system. They also can be easily extended while this will not imply the modification of a real physical process. On the other hand, while they virtualize the process and the control devices, they cannot reproduce the behavior of real field devices like Programmable Logic Controllers (PLC) or Intelligent Electronic Devices (IED) used in electrical networks. Therefore it is not possible to reproduce threats that exploit vulnerabilities of real devices.

Contributions: In this paper, we present two twin demonstrators based on the same technology: (i) an Advanced Persistent Threat (APT) demonstrator used for awareness training and (ii) a flexible lab used for students training and pentesting. Both are based on a common Hardware-In-the-Loop (HIL) technology which combines the advantages of virtualization and real cyber-ranges. We adopted a solution that only virtualizes the physical process alongside sensors and actuators. These virtualized components are connected with real industrial control devices in this HIL setup using an open source electronic interface system.

Outline: In Section 2, we briefly present the HIL system originally presented in [12]. Then, we present the APT training scenarios in Section 3, followed by the pen-testing, protocol fuzzing and reverse-engineering labs in Section 4. We conclude the paper with a description of the feedback we obtained from demonstrations and trainings and detail our future development plans.

2 HIL CYBER-RANGES (WONDERICS, G-ICS)

In this section, we present the Hardware-In-the-Loop technology shared by both cyber-ranges. These testbeds allow the reproduction of a complete and realistic industrial control system. The global architecture of these platforms is represented in Figure 1.

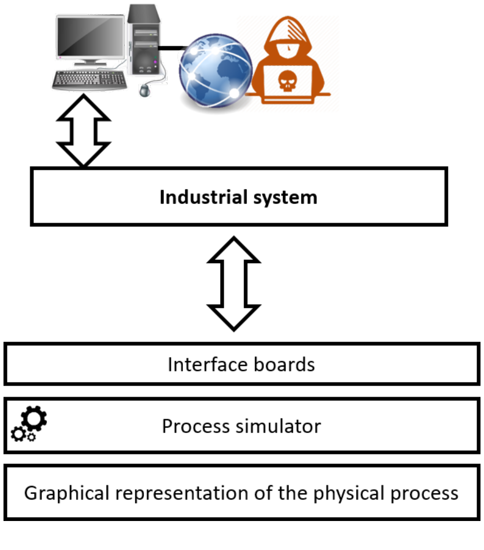

Figure 1: Overview of the presented platforms

To replicate correctly the behavior of an industrial systems, we aim to include all different layers of the Purdue model [17]. Thus, both platforms include real industrial devices and commercial SCADA software mixed with simulation for the physical process, its sensors and actuators. Because industrial interfaces from PLCs are not compatible with those of a computer, we developed interface electronic boards able to connect industrials devices to a computer running the simulations. We also intend to reproduce industrial systems of a size close to real industrial cases. Our symbolic target is one hundred industrial equipments and one thousand sensors and actuators. Thus, the interface board needs to scale up to the targeted number of inputs and outputs. Connected to these interface boards, the simulator of physical process is an open python-based software able to reproduce the behavior of different use-cases such as the management of hazardous gazes or the chemical process of Tenessee-Eastman [11]. An optional software is able to communicate with the simulator through a shared database to animate in real-time a graphical representation of the physical process. The rest of this section presents the common elements between the two testbeds: interface boards and the simulator.

2.1 Interface Boards

As explained in the above, our interface boards allow communication between the physical process simulator and off-the-shelf industrial hardware components from manufacturing and smart-grids applications fields. Figure 2 displays a synoptic of these boards, detailing their inputs and outputs.

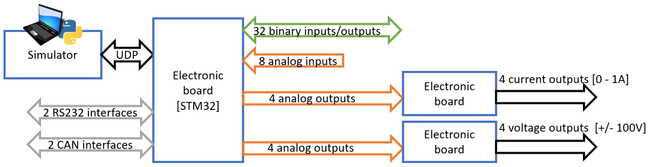

Figure 2: Interface board synoptic

These industrial devices can be Programmable Logic Controllers, Remote Terminal Units (RTU), Human Machine Interfaces (HMI) or embedded regulators and they interact with the physical process via a variety of I/O modules and interfaces. The electronic interface board integrates a quite large set of digital and analog inputs/outputs. As the electrical characteristics of the input and output signals vary a lot in industrial devices, we focused on the most common electrical signals (i.e. 0/24 V digital signals and -10/+10 V analogs). The current version of the interface board does not support specialized signals (pulse train or Pulse Width Modulation) or 4/20

[3] https://www.boeing.com/defense/cybersecurity-information-management/
[4] https://airbus-cyber-security.com/products-and-services/prevent/cyberrange/

mA current loops. The board also integrates industrial serial interfaces such as Modbus RTU and CAN. These legacy communication interfaces are still very used to drive actuators and sensors.

Smart-grid hardware controllers are mostly Intelligent Electronic Devices (IED) preprogrammed with electrical protection functions. To measure current and voltage in an electric network, these devices use digital and specialized analog inputs. Two normalized current levels are mainly used: 0/1 A and 0/5 A and voltage sensors are issuing signals in the range 0/100 V. For instance, protection relays are a type of IED commonly used in power management that measures and analyses the three phases of the electrical signals to detect and localize electrical faults such as over-current and control and electrical circuit breaker that will isolate the faulty circuit. Two other electronic boards are dedicated for current and voltage conversions and allow to emulate electrical signals compatible with IED. These boards can be used to demonstrate real attacks on electric networks. Figure 3 presents the two use cases of the interface cards.

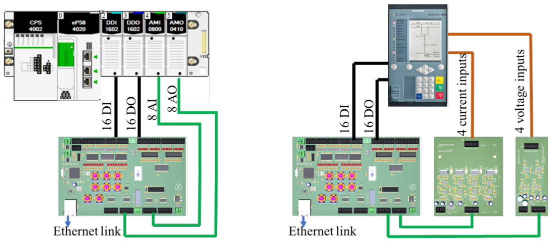

Figure 3: Interface board connection : PLC (left) and protection relay (right)

In terms of scalability, the use of an Ethernet network to connect interface boards to the simulator facilitates the integration of a new interface board allowing the implementation of real size industrial systems simulations. The number of connected interface board is only limited by the bandwidth of the Ethernet network.

In order to made the boards accessible to university labs we try to keep the price as low as possible. There is an obvious trade-off between price and performance. A professional high-precision testing hardware for IED may cost as high as 40.000€ for a single measurement point signal generator. We want to keep the price affordable then we fixed a target objective cost of 400€ per interface card and another 400€ for a couple of power cards (current transformer and voltage transformer).

2.2 Physical Process Simulation

The physical process simulator is a software reproducing an industrial physical process (e.g., a power plant). We designed this simulator with the goal to be easily adaptable to many use cases, thus being able to reproduce various processes. Moreover, we made it able to communicate with both generic IT and real off-the-shelf industrial components. For instance, the platforms we describe below include real industrial devices such as programmable automata. Thus, the simulator will be required to understand uncommon communication media such as electrical inputs or serial buses through interface boards presented in Section 2.1. It can also directly communicate with real components using industrial protocols over TCP/IP such as Modbus or OPC-UA. We propose a physical process simulator written in Python and based around the *simpy*[5] module. It is a process-based discrete-event simulation framework allowing us to model and schedule concurrently physical devices such as valves or coolers.

In our simulator, we chose to model such devices as independent tasks, scheduled on a shared clock. That is, each component is woken up at each – multiple of a – clock tick and can compute its outputs given its inputs. In other words, each component type (e.g., a valve) is modeled as a Python class implementing a special method called *process*. This method is the one called by *simpy* every tick, and processing inputs in order to compute outputs. This way, components can be reused in multiple physical process models as classes part of a library. In a main script, these classes are instantiated with their inputs specified as *lambda* functions (or Python properties). This construction allows the attributes of the modeled device to be physically the same variable as the one in the main script and not a copy passed by value. Then, the *process* method is registered in *simpy*'s scheduler alongside any other component or method needed by the model (e.g., input control or output monitoring). A working toy example is given for a valve in Listing 1 with the main script in Listing 2.

Multiple interfaces are built in the simulator to allow communications with other simulated or real components. First, we designed a UDP library handling a custom communication protocol with the interface board described in Section 2.1. This protocol allows to read digital (boolean) and analog (uint16) inputs and to respectively write outputs. Upon need, real industrial protocol servers can also be started inside the simulator to allow real components or devices to communicate with it. Obviously, as the simulator runs on a computer, a matching physical interface will be needed to handle connections (e.g., an Ethernet port for TCP/IP or a FTDI cable for serial protocols). The simulator also provides a RESTful API allowing to synchronize inputs and outputs with databases for data persistency. Moreover, this REST API potentially allows interaction with other programs such as a visualization of the values of the simulated components or direct modifications of variables with a presentation tablet. The process simulator's code is available on demand on a GIT repository[6].

```
class Valve():
    _opened     = None
    _inputFlow  = None
    _outputFlow = None

    def __init__(self, opened, inputFlow):
        self._opened    = opened
        self._inputFlow = inputFlow

    def process(self, env):
        while True:
            if self._opened():
                self._outputFlow = self._inputFlow()
            else:
                self._outputFlow = 0
            yield env.timeout(1)
```

Listing 1: A valve component

[5] https://simpy.readthedocs.io/
[6] https://gforge.inria.fr/projects/eastman/

```python
import  simpy

opened = True
inputFlow = 50
env = simpy.Environment()
///
def controlValve():
    global opened, inputFlow
    while True:
        # Code controlling the
        # valve during simulation
        yield env.timeout(1)

valve = Valve(
    env,
    opened=lambda: opened,
    inputFlow=lambda: inputFlow
)

env.process(controleValve())
env.process(valve.process(env))
env.run()
```

Listing 2: Main script

3 AWARENESS TRAINING AND APT DEMONSTRATORS

WonderICS is a hardware-software co-simulation environment mainly dedicated to raise awareness of cybersecurity issues in industrial control systems and to experiment innovative security solutions. This platform integrates the simulator presented in Section 2.2 that emulates physical processes based on three different use cases: hazardous gases management, hydroelectric power plant and the chemical process of Tennessee Eastman. A complete infrastructure including some virtual machines and a set of dedicated tools have been developed to attack the industrial control system in different ways (phishing mails, corrupted USB key, hardware trojan, etc). This section will describe the infrastructure allowing to attack the industrial system and some attacks developed especially for the different use cases.

Figure 4 shows a view from the WonderICS platform with, from left to right, we have a 3D view of the physical process, the SCADA software, off-the-shelf industrial devices, the computer of an operator targeted by attacks, and finally the attacker's computer. It is worth noting that the 3D view of the process represents the real state of the simulated process (and would be replaced by the actual physical process itself in a real factory) ; while the SCADA only show the local vision of the process obtain by communicating with captors and actuators.

3.1 Infrastructure of the WonderICS platform

To create realistic attacks that we can play on demand (and easily recover from), we have developed a complete network of virtual machines. These virtual machines allow us to implement complex attack scenarios with potentially high impact on the system but high isolation from hosts and seamless reset process. Figure 5 describes this virtual infrastructure.

Two physical computers host the different virtual machines. The attacker's computer is both used to launch attacks and to host attacker's downloadable resources (local mail server, website, etc).

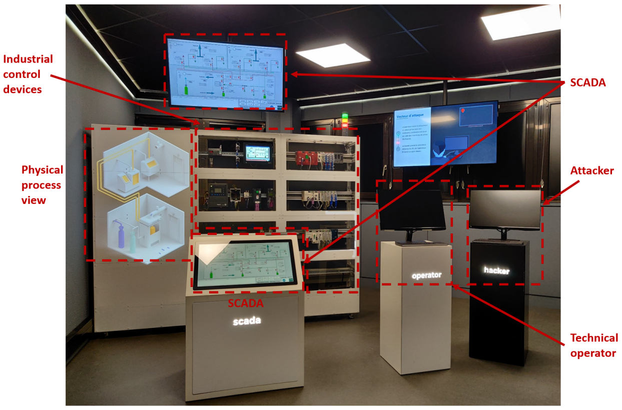

Figure 4: WonderICS platform

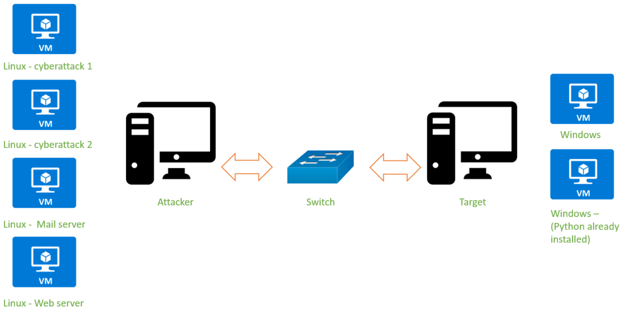

Figure 5: Network of virtual machines

The following virtual machines are implemented on the attacker's computer:

- Cyberattack 1: This virtual machine contains a Kali Linux operating system. The first cyberattack, explained in Section 3.2, is based on a Rubber Ducky key that emulates a keyboard. The tools needed for the first cyberattack have been added in this virtual machine. For example, the duckencoder program used to encode scripts on the Rubber Ducky key.
- Cyberattack 2: This machine also contains a Kali Linux OS but the APT attack materials are different. The second cyberattack, described in Section 3.2 integrates a RAT (Remote Access Trojan) server and mail client such as Mozilla Thunderbird to send some malicious emails as part of a spear phishing campaign.
- Mail server: This machine represents the local mail server of the industrial facility, allowing the attacker to send emails to the operator. The mail server is composed by a Postfix server for sending and a Dovecot IMAP server for receiving.
- Web server: A Nginx Web server is launched automatically at the start of this VM and allows to retrieve different programs part of attacks (charges, cryptolocker for example).

- Windows 7: This machine allows the use Window's Remote Desktop Control module that can give a complete access to the target's computer.

The majority of attacks target the computer of a technical operator of the industrial system because his computer has a direct access to the industrial network. The operating system used by the target is mainly Windows so we have chosen virtual machines running on Windows 7 operating system (which remains rather common in industrial facilities). These machines integrate Microsoft Office 2010 and Mozilla Thunderbird for spear phishing attacks.

3.2 Presentation of Attacks

As explained in Section 3.1, we implemented two complete cyberattacks reproducing the behavior of Advanced Persistent Threats (APT). Thus, they include several phases going from open-source intelligence to complete exploitation and include persistence. Global attack paths are represented in Figure 6. Both attacks start with a reconnaissance phase (1), then the delivery phase (2) allows the attacker to gain access to the vulnerable machine. Technically, at this step, both attacks allow for a control command server (3. C&C) to perform remote operation in order to gain privilege and/or information gathering. However, as detailed later, nature of Cyberattack 1 requires fast exploitation and would usually skip this phase. Both attacks then perform the actual charge delivery (4), harming the system and potentially the industrial process behind (5). The rest of the section details both cyberattacks (depicted in green and blue on Figure 6), then Section 3.3 will detail how these attacks can impact the industrial process controlled by the system.

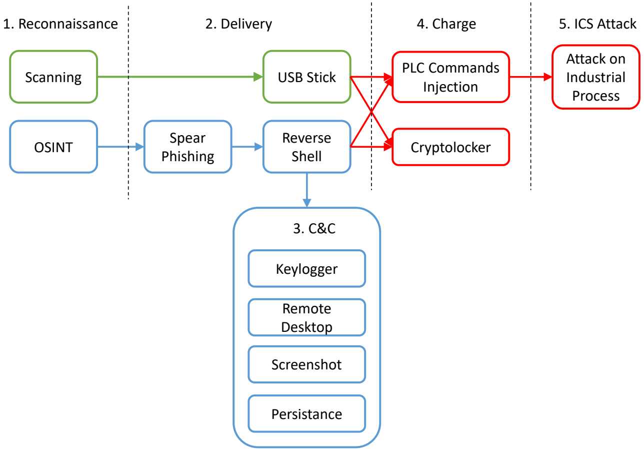

Figure 6: Attacks Steps

Presentation of Cyberattack 1: Cyberattack 1 involves the use of a Rubber Ducky[7]. The attack is based on the lack of precaution of an employee who, having found what looks like a USB key, inserts it into a USB port on his workstation. The Rubber Ducky is then considered as a keyboard, and programmed to perform a large number of actions at high speed. As the USB stick acts as a keyboard, all actions are visible on the operator's screen, leading to an immediate disclosure of the attack and requiring a quick charge delivery and exploitation. As this attack is completely remote and does not let much time for the attacker to perform manual operations, reconnaissance phase will mainly resume to scanning and information gathering to find vulnerabilities to exploit in the OS or applications. It then mainly relies on Python to execute actions and is able to disable other existing keyboards to prevent the operator stopping the attack. It will also be able to fetch a malware from a remote server if needed. As the charge is launched without any control from the attacker, it is usually more realistic to conclude this attack with automatically launching a cryptolocker. An overview of cyberattack 1 is depicted in Figure 7. In an awareness demonstration, this cyberattack is very visual and impressive for decision-makers as a lot of actions are quickly performed by the USB stick and they loose the control of the target system.

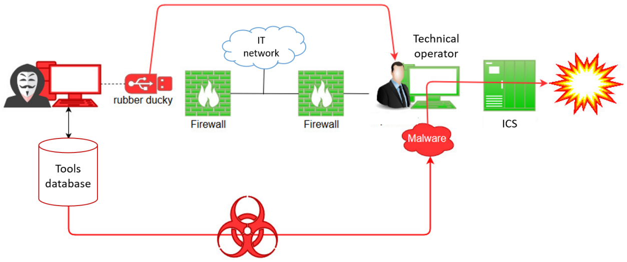

Figure 7: Overview of attack 1

Presentation of Cyberattack 2: Cyberattack 2 involves sending a fake email containing a Word document with a rogue macro that sets up a Remote Access Trojan (RAT) on the Target PC. Once this RAT is set up, the attacker can initiate the download of charges from a controlled web server to the Target machine, and launch their execution. This attack has a lot from Advanced Persistent Threats and starts with Open-Source Intelligence (OSINT) in order to gather information on a vulnerable employee (his work email, his position, his hobbies, etc). It then relies on spear phishing with an email specifically destined to him and designed around his involvement in the associative world to maximize his chances to open the malicious office document attached. This document starts a RAT in the background that will automatically connect the a Control Command server (C&C) setup by the attacker and allow him to use various commands on the target system. These commands include launching a keylogger, enabling Windows remote desktop, taking a screenshot from the screen or making the malware persistent in case of reboot via various vulnerability exploits. Once the attacker estimates that he gathered enough knowledge on the system, he can download and run various charges in order to affect the industrial system behind the target system. These industrial focused charges are described in Section 3.3. An overview of cyberattack 2 is depicted in Figure 8.

3.3 Modbus Injection and Impact on the Physical Process

As soon as the attacker takes control of the operator's computer, he obtains direct access to industrial controllers (PLCs) and the

[7] https://shop.hak5.org/products/usb-rubber-ducky-deluxe

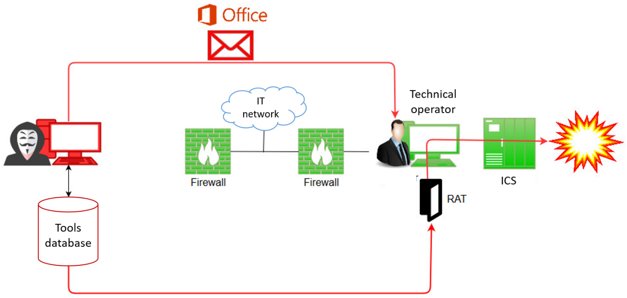

Figure 8: Overview of Attack 2

SCADA. In our configuration, the protocol Modbus TCP is used between the SCADA, the targeted computer and the Programmable Logic Controller (PLC). This protocol is legacy but still used in most of industrial systems. As in the majority of industrial protocols, there are no builtin security and a lot of vulnerabilities can be used to eavesdrop or inject false commands. Within the WonderICS platform, we consider two main types of attacks: (i) injection attacks, and (ii) man-in-the-middle attacks.

Injection attacks: Depending on the use case chosen in the WonderICS platform (hazardous gas management, hydroelectric plant, chemical process), we can inject Modbus frames to change the status of a sensor or send a new order to an actuator. Targeted devices of this attack are mainly controllers but the final target can also be a simulated device (valve, sensor, etc) or a real physical actuator connected to the controllers (protection relay, breaker, motor management, etc). For instance, one of our attack vector sends a malicious Modbus frame to write a fake value on the register address of the controller. This address encodes the value of the intensity given by an electrical protection relay. By sending a very high value of intensity, the controller detects an overcurrent and opens a real industrial breaker connected to the platform in order to cut the current in the system.

Man-in-the-middle attacks: While injection attacks don't require any actions from a legitimate client, Man-in-the-middle attacks aim at replacing the contents of Modbus frames as they are being sent by the SCADA. In this direction, it doesn't differ that much from injection attacks. However, by replacing the response of PLCs to requests sent by the SCADA, we manage to decorrelate the state of the process seen on the SCADA with the read state of the process shown in the 3D representation. For instance, in the hydroelectric power-plant scenario, we could cause the dam to overflow and hide all alarms to the SCADA.

3.4 An example of awareness training in WonderICS

As soon as they enters in the WonderICS platform, visitors are immediately immersed in the context of industrial cybersecurity due to the presence of real off-the-shelf devices and real visualization softwares. The demonstration is always calibrated to the visitor's knowledge and skills in cybersecurity. Thus, before explaining a cyberattack, the presenter explains them the architecture of the whole platform. A tablet used by the presenter allows him to project pictures on the electrical cabinet to present the different industrial networks or physical devices. Then, the presenter describes the use case; for example the management of an hydro power plant. Using the SCADA software and the projection of a graphical view of the physical process, he explains the different components of the physical process (dam, valves, turbo-alternator, protection relay, etc). Then, at the start-up of the scenario, the visitor sees on the graphical view the given facility running smoothly (e.g., the flow of water in the penstock or the generation of electricity in the generator). After this, the presenter introduces the context of the attack using some slides depending on the use case and the presented attack (e.g., the company that manages the hydro plant targeted by the attacker, the context of the attack and the profile of the attacker). The first step of attack demonstration shows simple ways for an attacker to perform OSINT and gather information about the targeted company and its employees. His objective is to find an employee whose position allows him to access the industrial system and whom he can corrupt or use his weaknesses to carry out his attack. In our usage scenario, the attacker mainly uses social networks to obtain personal information about the technical operator of the industrial system. Once the attacker obtained enough information on the target, he can forge an email or position a corrupted key on a place to set up the attacks presented in Section 3.2.

The objective of this kind of training is to understand the vulnerabilities used by an attacker to develop his attacks. Even if the attacks just last few minutes during the demonstration, it is important to explain to visitors that this kind of attack can require few months or years to be developed. Moreover, an important knowledge of the industrial system and physical process is needed, it may also be needed to corrupt the company's staff to obtain some secret information. Next, the attack is carried away as described in Section 3.2.

At the end of this demonstration, some security guidelines based on the rules of the French ANSSI (*Agence National de la Sécurité des Systèmes d'Information*), the french organization for computer security [4] [5] and the NIST (National Institute of Standards and Technology) [15] in relation with the aforementioned attacks are discussed. These technical and organizational principles are basic rules to improve the security of industrial systems.

4 PENTESTING AND REVERSE ENGINEERING FLEXIBLE TRAINING

G-ICS lab (GreEn-ER[8] Industrial Control systems Sandbox) is an industrial control systems research and teaching lab. More than 100 industrial devices (controllers, protection relays, remote terminal units and industrial HMI's) from several vendors are available together with several supervisory control servers and a few security devices (Stormshield firewalls and Cisco CyberVision IDS). A partial view or the lab is presented in Figure 9.

The lab is used both for research and training. On the research side, the lab was used for the experimental part of two Ph.D. theses

[8] Grenoble énergie - enseignement et recherche - an energy research and training center in Grenoble

Figure 9: G-ICS lab

on intrusion detection and as a demonstrator for the Grenoble Cybersecurity Institute[9] project. It has also provided a demonstration stand at International Cybersecurity Forum (FIC) at Lille since 2017. On the training side, the lab is used for electrical engineers training in industrial automation and supervisory control, industrial real-time communications and cybersecurity awareness of electrical engineering students but also for the computer science students in Cybersecurity Master major. In the following, we'll shortly present the two education programs.

4.1 Electrical Engineering Training Program

For electrical engineering students, the main objective is to acquire PLC programming and supervisory control skills. However, modern control engineering relies heavily on communication networks, thus a basic knowledge of industrial systems cybersecurity is mandatory. Our electrical engineering students are enrolled for an awareness training in cybersecurity compliant with the French CyberEDU[10] educational label. Therefore, their curricula is a mix of SCADA and cybersecurity training (Table 2).

Topic	Type	Duration
Computer networks	Master class	1 day
Industrial protocols	Master Class	1 day
Cybersecurity Primer	Master Class	1 day
SCADA	Lab	2 days
Industrial protocols	Lab	3 days
Basic security controls	Lab	2 days

Table 2: Electrical engineering track.

The master classes address the basics of TCP/IP protocols, fieldbus and industrial real-time protocols and an introduction to cybersecurity with an emphasis on industrial control systems: review of classical attacks and incidents concerning SCADA systems, main differences between IT and OT, common vulnerabilities of industrial devices and basic security controls. SCADA and Industrial protocols labs focuses not only on controller programming and SCADA design but also on traffic observation, network calculations and optimization. For a typical lab, we use a "serious game" plant simulation like Home I/O or Factory I/O from Real Games[11] as they are designed for education [13]. We interfaced the simulators with our HIL system and real PLCs using the Real Games API and the programs are available on-line[12].

In Figure 10, a screen capture shows the final set-up of a student lab: the simulated physical process (Home I/O screen lower right), control program running o the real controller (Schneider Control Expert screen upper left corner), SCADA screen connected with the physical controller (PCVue synoptic lower left corner) and the Wireshark network monitoring of the Modbus/TCP exchanges (upper right corner).

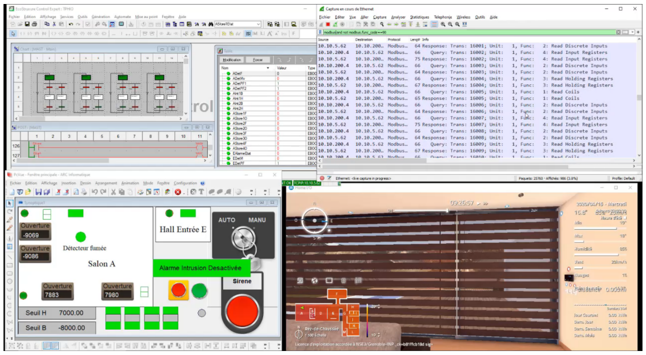

Figure 10: Plant/Controller/SCADA/Network traffic view

Network optimization lab is based on the analysis of the number of protocol requests versus the number and type of process variables read and write by the SCADA. It implies an organization of the PLC internal variables in contiguous blocks grouped by type (read-only or read-write) and polling period but also the programming of the SCADA client such that the number of requests is minimized. Finally, the cybersecurity lab implies the deployment of the available security controls. The considered attack scenarios are:

- false data injection into the PLC by a threat present on the SCADA network;
- PLC program modification via an unauthorized developer access;
- PLC internal data modification using an unsecured service (FTP, or embedded web site with default credentials).

PLC security controls: Students will consult the manufacturer specifications and establish the list of available controls then justify for each control the necessity and deployment level. The most common available controls are :

- Whitelist: a list of allowed IP addresses to connect and exchange data with a controller can be established. As only the IP address is verified this security control can be easily

[9] https://cybersecurity.univ-grenoble-alpes.fr/
[10] https://www.cyberedu.fr/
[11] https://realgames.co/
[12] http://lig-g-ics.imag.fr

avoided and the students are aware to never rely on whitelist only;

- Protected memory variables: exchanges between SCADA and PLC are using directly the PLC memory variables. On some PLC models a variable protection can be applied. Students have to list the used PLC memory addresses in the SCADA communication, check that they are organized by category in the PLC memory (read and read-write distinct blocks) then set-up the adequate access rights in the PLC configuration;
- Password protection of the PLC control program: By default programming access to the PLC is not password protected. Practically all recent programming environments allow password protection of the control program and raise a warning when password is absent.
- Shutting down unsecured services and securing the needed services: traditionally, control devices are using unsecured services like FTP, SNMP, SMTP and HTTP for non-real time communications like firmware upgrade, diagnostics and network and device monitoring. They are still available even on the last generation models but, hopefully, no more activated by default or, at least, a warning is raised by the development environment if an unsecured service is activated. Students have to list the available services, check the utility of each service for their application, set-up a secured version where possible (use of HTTPS instead of HTTP and SNMPv3 instead of SNMPv2, for instance) check that default login/passwords are not used, shut down unnecessary services (usually FTP) and list the remaining unsecured services (typically NTP and SMTP clients). A network protection mechanism will be required for the unsecured remaining services.
- Syslog setup: Some recent PLC models implement a syslog service and are even able to log messages on a remote syslog server. Students have to check the presence of the syslog service and set-up the client.

Network security controls: As the training addresses electrical engineering students, we do not ask them to do advanced tasks like setting up secure gateways and VPN channels. The labs are targeting the network monitoring and basic understanding of firewall rules. We are using CISCO Cisco Cyber Vision[13] for industrial traffic monitoring and alert raising. Using a cartography of normal traffic students shall propose firewall rules to block potentially abnormal requests. Using deep packet inspection capability of industrial firewalls (we use the Stormshield SNi40[14] in particular), one can detail the analysis down to Modbus/TCP function identifiers and memory addresses accessed.

Most off the shelf SCADA software will not provide the full lists of requests and protocols used in the implementation. Usually, one has to observe the traffic in order to decide which subset of requests of a given protocol are used in a given application. A critical analysis is used to define the limits of the monitoring based flow cartography while a firewall ruling, allowing only observed request may exclude and, therefore, it may block "rare" but legitimate requests not observed in the regular traffic (like alerts, for instance or device configuration requests).

4.2 Cybersecurity Master Training Program

A second track was set-up for cybersecurity students enrolled to a specialized Master program. While their future jobs will be related either to information security systems deployment and management or to cybersecurity controls development and device testing, the track is a mix between normative document study, risk analysis and pentesting.

Topic	Type	Duration
Cybersecurity of Industrial Systems	Master Class	0.5 day
Hands-on SCADA	Lab Tutorial	2 days
ISO/IEC 27000	Master class	1 day
EBIOS for Industrial systems	Master Class	1 day
STRIDE		0.5 day
IEC 62443	Master Class	1 day
Pentesting SCADA	Lab	1 days
Reverse engineering	Lab	1 days

Table 3: Cybersecurity Master track.

The first master class introduces the main concepts of SCADA systems and their cybersecurity. We present the basic notions of control systems, PLC and supervisory control and we focus on the industrial real-time aspects and their consequences to the device and communication design. Cybersecurity is introduced via classical examples of cyber-incidents and attacks (2006 U.S. blackout started by an accidental false-data injection, Stuxnet and Black Energy) we present the main differences between IT and OT systems from the point of view of cybersecurity and a we briefly introduce the standards that apply.

The "hands-on SCADA tutorial" is a fast initiation to PLC programming and supervisory control for non-control-system students. The trainees have to follow an example and set-up a basic SCADA system (one PLC and one supervision screen). The secondary objective is to let students understand the close link between the communication protocols and the control of the system. This point will be exploited in the pentesting labs and the studied use-case (Figure 11) will be used for cybersecurity master class applications. The physical process is a mixer of two products. The PLC program has to control the filling of the tank with the two products, control the mixer motor and the evacuation of final product. Only three sensors (level sensors E, P1 and P2) and four actuators (filling valves VP1 and VP2, the motor M and the flush valve VE) have to be controlled. The SCADA screen will visualize the states of the sensors and actuators and allow manual operations.

The next three master classes are intended to provide a comprehensive description of the SCADA cybersecurity and the state of the standardization. The IEC 27000 [7] class covers the essentials of the standard (management system, risk management, security controls) and also sector specific information, in particular for energy (ISO 27019). A simple risk-analysis and security controls deployment exercise is conducted on the use-case from the SCADA tutorial.

[13] https://www.cisco.com/c/en/us/products/security/cyber-vision/index.html
[14] https://www.stormshield.com/products/sni40/

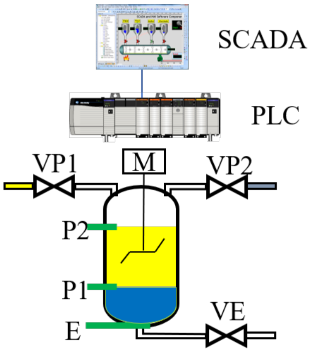

Figure 11: Simple use-case for SCADA tutorial and pentesting

Two risk analysis methods are presented. The EBIOS[15] for industrial systems master class is a one day tutorial presenting the French risk management system applied to industrial systems case. We conduct an interactive EBIOS exercise on the studied use-case. STRIDE is a system threat modeling technique based on the analysis of an information flow diagram. There are six threat categories considered and the STRIDE acronym is a mnemonic of the threat names: Spoofing, Tampering, Repudiation, Information disclosure, Denial of service and Elevation of privilege. The modeling technique uses a graphical representation of the systems based on Data Flow Diagrams (DFD) enriched with confidence boundaries which are sets of entities (processes, flows and data storage of same privilege level). The method was initiated by Microsoft and spread especially in the industrial world. We use the Microsoft Threat Modeling Tool[16] to model the communication of simple use-case as an application. One of the automatically generated threat model will be implemented in the lab as a real attack.

IEC 62443 [3] is the newest cybersecurity standard for industrial systems. It was originally initiated by International Society of Automation and some parts are still under development. Several certifications schemes are already defined and all major device manufacturers are certifying their equipments. The 62443 approach redefines the security objectives in a more "control system" manner and proposes a partitioning of the systems in security zones. The idea is that, in a real control system, the security needs and the capability of the devices are not the same at different levels. For instance, it will be very difficult to encrypt communication between sensor and actuators at field level while keeping the real-time performance. Therefore, the 62443 aims to group devices with the same requirements and capabilities into zones and to control the communication between zones. Thus, the security approach is mostly related to network segmentation. We conduct a network segmentation exercise on firstly on the simple use-case then on a complex. The trainees are required to identify zones and flows between zones then write the fire-wall rules for the corresponding network segmentation.

The pentesting SCADA lab contains two parts: the first one consists in a false data injection attack implementation and test on real devices. The students are asked to study the traffic between SCADA and PLC in simple use-cases and identify the network frames used to remotely control the actuators then write a program that will inject malicious controls into the systems and compromise the process (for example open the flush while the motor is still running or continue to fill the tank after the nominal level is attained). The effects of the attack are visible on the simulated physical process. The second part concerns the test of some known exploits. For instance we test the CVE-2013-2763[17], as the exploit can be found on internet. After testing and visualizing the exploit the students are required to write a firewall rule for blocking the known exploit.

The protocol reverse engineering lab continues the pentesting lab with simple illustration of techniques employed to find vulnerabilities and exploits. We study the UMAS[18] protocol whose specifications are not public. For the reverse engineering part, we use a traffic capture between an HMI and a PLC corresponding to a known operation (for instance forcing an output on a PLC) students are required to find the protocols field used to specify the address of the output then use this partial information to build an attack and finally implement and test the attack. Protocol fuzzing is used to detects some legal UMAS requests and even to found the exploit associated to CVE-2013-2763.

5 CONCLUSION AND FURTHER DEVELOPMENTS

In this paper, we presented two twin demonstrators based on the same technology: (i) an Advanced Persistent Threat (APT) demonstrator used for awareness training (WonderICS) and (ii) a flexible lab used for students training and pentesting (G-ICS). Both are based on a common Hardware-In-the-Loop technology where virtualized components reproducing the physical process are connected with real industrial control devices.

The two training programs using G-ICS lab started in 2016 (for electrical engineering students) respectively in 2018 for cybersecurity Master students. As the ICS cybersecurity is still an "exotic" field and the hacking part is quite spectacular both electrical engineering and cybersecurity students declare to be delighted. Let alone the students enthusiasm for exploit and hacking, a detailed interview showed that, for electrical engineers the training demystified the cybersecurity and some of them even decides to redirect their carrier (around 2%). Cybersecurity master students declared that the training was useful for them as they discovered the control system domain. As the job offer in industrial control systems security is growing the students declare to feel for confident for applying. The WonderICS platform was used in multiple demonstrations for industrial vendors and stakeholders. All of them expressed a lot of interest for the demos and the underlying problem of the cybersecurity of industrial systems. Most visitors were also quite shocked by the simplicity and the lack of requirements to launch a cyberattack on an ICS. They were also keen on seeing how an APT is construct and how it can present being detected.

[15] Expression des Besoins et Identification des Objectifs de Sécurité : Expression of need and security requirements identification in French

[16] https://www.microsoft.com/en-us/securityengineering/sdl/threatmodeling

[17] https://nvd.nist.gov/vuln/detail/CVE-2013-2763

[18] https://documentation.stormshield.eu/SNS/v4/en/Content/User_Configuration_Manual_SNS_v4/Protocols/SCADA-UMAS.htm

In the future we intend to develop red team/blue team games on our labs and we think at easily reproducible scenarios. Another possibility would be the integration with existing automated platforms such that the newly released Microsoft CyberBattleSim [16]. For the moment, CyberBattleSim do not provide an interface with real word neither a support for industrial systems, but openness of the project and the presence of an development environment makes it an interesting solution. We also plan to extend supported devices of both platforms either by integrating IIoT devices communicating over wireless media (e.g., ZigBee, Bluetooth) or to allow WonderICS to communicate with emulated devices via the WonderCloud infrastructure [8] developed alongside WonderICS.

REFERENCES

[1] Giddeon N. Angafor, Iryna Yevseyeva, and Ying He. 2020. Bridging the Cyber Security Skills Gap: Using Tabletop Exercises to Solve the CSSG Crisis. In *Serious Games*, Minhua Ma, Bobbie Fletcher, Stefan Göbel, Jannicke Baalsrud Hauge, and Tim Marsh (Eds.). Springer International Publishing, Cham, 117–131.

[2] BSI. 2019. *Top 10 Threats and Countermeasures.* RECOMMENDATION: IT IN PRODUCTION. Federal Office for Information Security. https://www.allianz-fuer-cybersicherheit.de/SharedDocs/Downloads/Webs/ACS/DE/BSI-CS/BSI-CS_005E.pdf

[3] International Electrotechnical Commission. 2009. *IEC/TS 62443-1-1:2009 Industrial communication networks – Network and system security – Part 1-1: Terminology, concepts and models*. 62443, Vol. 6. Afnor.

[4] Agence Nationale de la Sécurité des Systèmes d'Information. [n.d.]. *La cybersécurité des systèmes industriels: Maîtriser la SSI pour les systèmes industriels*. ANSSI. https://www.ssi.gouv.fr/uploads/IMG/pdf/Guide_securite_industrielle_Version_finale.pdf

[5] Agence Nationale de la Sécurité des Systèmes d'Information. [n.d.]. *La cybersécurité des systèmes industriels: Mesures détaillées*. ANSSI. https://www.ssi.gouv.fr/uploads/2014/01/securite_industrielle_GT_details_principales_mesures.pdf

[6] European Parliament and Council of European Union. 2016. Directive (EU) no 2016/1148 concerning measures for a high common level of security of network and information systems across the Union. https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32016L1148&from=EN http://eur-lex.europa.eu/legal-content/EN/TXT/?qid=1416170084502&uri=CELEX:32014R0269.

[7] International Organization for Standardization. 2018. *ISO/IEC 27000:2018: Information technology — Security techniques — Information security management systems — Overview and vocabulary.* ISO/IEC, Geneva. https://www.iso.org/standard/73906.html

[8] Mathieu Gallissot, Maxime Puys, and Pierre-Henri Thevenon. 2020. WonderCloud, une plateforme pour l'analyse et l'émulation de micrologiciels ainsi que la composition de pots de miels. In *C&esar 2020 - Deceptive Security*. Rennes, France.

[9] INL. 2008. *Common Cyber Security Vulnerabilities Observed in Control Systems Assessments by INL NSTB Program.* report. Idaho national Laboratory.

[10] Ralph Langner. 2011. Stuxnet: Dissecting a cyberwarfare weapon. *IEEE Security & Privacy* 9, 3 (2011), 49–51.

[11] TJ McAvoy and Nan Ye. 1994. Base control for the Tennessee Eastman problem. *Computers & Chemical Engineering* 18, 5 (1994), 383–413.

[12] Stéphane Mocanu, Maxime Puys, and Pierre-Henri Thevenon. 2019. An Open-Source Hardware-In-The-Loop Virtualization System for Cybersecurity Studies of SCADA Systems. In *C&esar 2019 - Virtualization and Cybersecurity*. Rennes, France, 1–16. https://hal.archives-ouvertes.fr/hal-02371133

[13] B. Riera and B. Vigário. 2017. HOME I/O and FACTORY I/O: a virtual house and a virtual plant for control education. *IFAC-PapersOnLine* 50, 1 (2017), 9144–9149. https://doi.org/10.1016/j.ifacol.2017.08.1719 20th IFAC World Congress.

[14] Elena Sitnikova, Ernest Foo, and Rayford B. Vaughn. 2013. The Power of Hands-On Exercises in SCADA Cyber Security Education. In *Information Assurance and Security Education and Training*, Ronald C. Dodge and Lynn Futcher (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 83–94.

[15] Keith Stouffer, Victoria Pillitteri, Suzanne Lightman, Marshall Abrams, and Adam Hahn. 2015. *NIST Special Publication 800-82 - Guide to Industrial Control Systems (ICS) Security.* NIST. https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-82r2.pdf

[16] Microsoft Defender Research Team. 2021. CyberBattleSim. https://github.com/microsoft/cyberbattlesim. Created by Christian Seifert, Michael Betser, William Blum, James Bono, Kate Farris, Emily Goren, Justin Grana, Kristian Holsheimer, Brandon Marken, Joshua Neil, Nicole Nichols, Jugal Parikh, Haoran Wei.

[17] Theodore J Williams. 1989. A reference model for computer integrated manufacturing (CIM). *International Purdue Works* (1989).