Employing Digital Twins for Security-by-Design System Testing

Marietheres Dietz
marietheres.dietz@ur.de
Department of Information Systems
University of Regensburg
Germany

Leon Hagemann
leon-david.hagemann@stud.uni-regensburg.de
University of Regensburg
Germany

Constantin von Hornung
constantin.von-hornung@stud.uni-regensburg.de
University of Regensburg
Germany

Günther Pernul
guenther.pernul@ur.de
Department of Information Systems
University of Regensburg
Germany

ABSTRACT

Ever since cyber attacks focused on industrial and critical infrastructure settings, the awareness of the security issues of these systems has increased. These industrial control systems (ICS) mainly focus on operation and availability – instead of providing general security features. Moreover, the current Industry 4.0 movement aggravates this security gap by connecting the ICS to the enterprise network, which facilitates targeting these systems. Proper system testing can reveal the system's vulnerabilities and provide remedies. However, security measures are usually neglected or addressed after an emerging incident only, which results in high costs. To maximize the benefit of system testing, we argue that it should be carried out as early as possible, especially to render systems secure-by-design. In this work, we propose an approach for introducing security-by-design system testing by the application of a digital twin. A digital twin is able to represent a system virtually along its lifecycle. To enable security-by-design, the simulation capability of digital twin is harnessed to create a prospective environment of a planned system. This allows detecting vulnerabilities before they can emerge in the real-world and providing a adequate risk strategy. Our work shows how security-by-design system testing is anchored in the security applications along a system's lifecycle. Next to proposing a security-by-design system testing approach with digital twins, we implement a digital twin representing a pressure vessel, and demonstrate how to carry out each step of our proposed approach. During this proof-of-concept, we identify vulnerabilities and show how an attacker can compromise the system by manipulating values of the pressure vessel with the potential to cause over-pressure, which, in turn, can result in an explosion of the vessel.

CCS CONCEPTS

• **Security and privacy** → **Vulnerability management**; • **Computer systems organization** → Embedded and cyber-physical systems; • **Networks** → *Network protocols.*

SaT-CPS '22, April 27, 2022, Baltimore, MD, USA

ACM ISBN 978-1-4503-9229-7/22/04.
https://doi.org/10.1145/3510547.3517929

KEYWORDS

digital twin, system testing, security-by-design, industrial vulnerabilities

ACM Reference Format:
Marietheres Dietz, Leon Hagemann, Constantin von Hornung, and Günther Pernul. 2022. Employing Digital Twins for Security-by-Design System Testing. In *Proceedings of the 2022 ACM Workshop on Secure and Trustworthy Cyber-Physical Systems (SaT-CPS '22), April 27, 2022, Baltimore, MD, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3510547.3517929

1 INTRODUCTION

At first glance, industrial systems might resemble enterprise systems. Nevertheless, there is one great difference between these systems. While enterprise systems focus on information, they emphasize confidentiality. Systems deployed at industrial sites, however, put their focus strongly on its operation and thus, its availability. While this might seem a small remark, it has turned into a problem ever since attacks targeted industrial and critical infrastructure environments (e.g., the Stuxnet incident [17] or the recent industrial attack Triton [19]). Besides, along the movement of Industry 4.0, enterprise information technology (IT) systems and industrial operational technology (OT) systems increasingly converge [27]. This results in phenomena like connecting the industrial control systems (ICS) to enterprise networks, which further enlarges the attack surface of the often insufficiently secured ICS.

To increase security, testing systems in order to reveal potential vulnerabilities and provide mitigation is a common procedure. In ICS, system security testing remains problematic [15] as the 24/7-availability of the system must be ensured. An outage or disruption of the operation might produce high costs that can even threaten the existence of the industrial corporation. Therefore, the functioning of the industrial operation predominates security, although security is desperately required to ensure the proper functioning without potential harms from malicious activities. This vicious cycle explains why serious security concerns commonly arise in ICS and critical infrastructures. In most industrial systems, security is addressed no earlier than after an incident occurs. Thus, throughout the ICS's lifecycle, security is tackled in medium to late phases, which take place during or after the system's operation.

A potential solution to this severe problem would be to implement security in earlier lifecycle phases (e.g., during planning, designing or engineering of a system). This strategy is known as security-by-design. It however entails that system testing cannot be conducted in the real-world system as the system has not yet physically manifested. Thus, to enable system testing jointly with security-by-design, recourse has to be taken to virtual representations of the would-be systems.

Employing a digital twin to this purpose may facilitate security-by-design system testing. The digital twin is generally referred to as a virtual representation of a system during its entire lifecycle. Thereby, it can exist before the system is built [5] and might support the security-by-design approach during the development of an ICS. This might include using the simulation capability of the digital twin, which can create a prospective environment and allow for system testing to detect vulnerabilities.

In this work, we propose to conduct security-by-design system testing with the digital twin. To this end, we first categorize digital twin security applications to the corresponding system's lifecycle phases. From there, we focus on the security-by-design system testing, deduce conceptual preliminaries and develop our main approach. Finally, we give an exemplary demonstration of conducting security-by-design testing with a digital twin setting representing a pressure vessel. The contributions of our work can be summarized as follows:

- provide an overview of digital twin security applications along a system's lifecycle
- propose a concept for security-by-design system testing with the digital twin
- present proof-of-concept by digital twin implementation and use case

The remainder is structured in six parts. At first, the background is laid and the related works are presented in Section 2. The main concept with its preliminaries is described subsequently (Section 3). In order to provide proof-of-concept, a digital twin of a pressure vessel is implemented in Section 4. Afterwards, the proposed security testing approach is carried out with the implemented digital twin use case (Section 5). Various aspects of this research are critically discussed in Section 6. A conclusion is finally drawn in Section 7.

2 BACKGROUND AND RELATED WORK

To lay the scientific foundations for the proposed approach, Section 2.1 introduces the concept of the digital twin and illustrates its connection to security. Section 2.2 provides the background to security testing and the security-by-design paradigm. To conclude, we address related works in Section 2.3.

2.1 Digital twin

The digital twin is a difficult term with various definitions – depending on its application domain [23]. Most generally, the digital twin presents its real-world counterpart virtually and throughout its lifecycle. Thus, the digital twin paradigm commonly comprises the digital twin (a virtual solution) and the real-world counterpart, which is often referred to as physical twin [30]. However, owing to the aspect of *accompanying its counterpart's lifecycle*, a digital twin can exist even before its counterparts manifests physically [5]. Consider for instance a company planning the manufacture of a new windmill. First steps are planning the mill's composition, including the subparts and their collaboration in order to ensure its proper operation. To this end, the digital twin could be utilized, and virtually present the planned windmill, although the mill has not yet been manufactured (physically manifested).

Manifold digital twin applications exist that can be mapped to the counterpart's lifecycle [10]: Early lifecycle phases (e.g., engineering) include designing, system testing, and virtual commissioning with the digital twin. At the operation phase, the twin is mainly used for maintenance. At the end-of-life, decommissioning is a main use case. For adequate representation, the digital twin employs simulations and other intelligent services, and relies on semantically linked data about its counterpart [3].

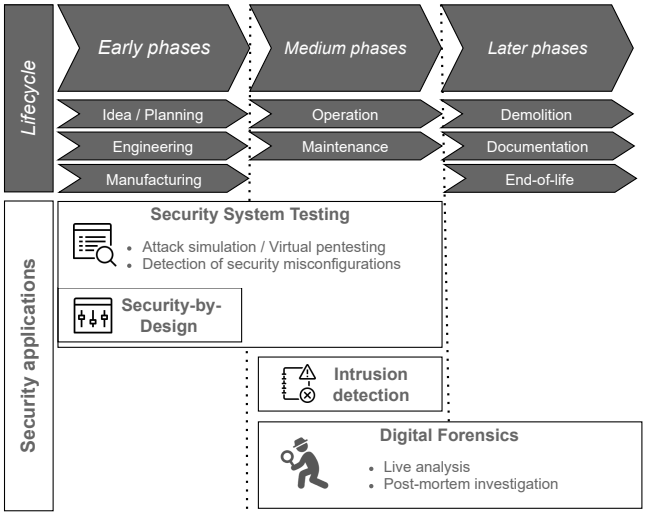

Figure 1: Digital twin security applications mapped to its counterpart's lifecycle.

The digital twin is also linked to security. While literature usually mentions proper security a major requirement for digital twin deployment [14, 32], recent works focus on harnessing digital twins for security purposes [10, 27]. Four use cases applying digital twins for security are proposed [9]: intrusion detection, system testing & simulation, detection of misconfigurations and penetration testing. Moreover, a multitude of benefits for employing digital twins in security are introduced [6], including rendering systems secure-by-design, applying digital twins for digital forensics as well as conducting safety and security co-analysis with digital twins. Figure 1 summarizes the digital twin security applications in respect to the counterpart's lifecycle based on the works of [6, 9, 10]. This work focuses on applying system testing with the digital twin to enable secure-by-design systems, which can be located at early lifecycle phases (see Figure 1).

2.2 Security-by-design system testing

System testing is used for manifold purposes. It can be applied to measure functionality and to evaluate performance, but also to test the system's security state. System security testing can be conducted by following frameworks like the Information Systems

Security Assessment Framework (ISSAF). Furthermore, alternative system security frameworks are proposed in literature (e.g, [12, 26]). Official guides like the National Institute of Standards and Technology (NIST) 800-115 technical guide to security testing and assessment [28] or the Open Source Security Testing Methodology Manual (OSSTMM) also propose guidance for carrying out security tests.

The *security-by-design* paradigm intends to implement proper security measures before a system is built and before it takes on operational tasks [29], i.e. in early lifecycle phases (esp. design phase). By applying the paradigm, the security and cyber defence costs can be minimized [6]: If security mechanisms are installed at the beginning, the system will be less vulnerable to cyber attacks in the future. As a result, the tremendous sums for attack recovery and implementing security mechanisms in hindsight, which amount usually to higher costs than implementations in the beginning, are saved.

In order to conduct security-by-design system testing, a novel approach is necessary. As in early lifecycle phases no real-world system is physically available, and therefore cannot be tested, the security testing has to be carried out in a virtual environment representing the currently designed system.

2.3 Related works

Related works also show that early testing of an ICS environment is commonly conducted in virtual environments. On the one hand, *virtual testbeds* are used. For instance, a virtual water supply system testbed is created and an attack on the communication is carried out in [31]. Another digital testbed of a water supply system detecting an attack is presented in [22].

On the other hand, *digital twins* can be applied as virtual environments. A digital twin, created from the program code of an existing physical plant, is used to find novel attacks and vulnerabilities in [13]. Implementing the CPS Twinning framework to create a digital twin, it is shown how an attack can be detected in [9]. Another digital twin that can detect attacks on an ICS is shown in [4]. The integration of a digital twin with a Security Information and Event Management system (SIEM) is described and prototypically implemented in [7].

A first conceptual approach of security system testing with digital twins is recently introduced [2]. However, this short proposition remains theoretical and is limited to the area of smart grids, while our work refers to digital twins of any ICS and provides an exemplary digital twin implementation of an industrial setting (pressure vessel). We further show how digital twins can support the security-by-design concept.

3 DIGITAL TWIN-BASED APPROACH FOR SECURITY-BY-DESIGN TESTING

In order to derive a sound approach for security-by-design system testing with the digital twin, preliminaries are set in Section 3.1. On this basis, Section 3.2 describes the proposed approach.

3.1 Conceptual preliminaries

To perform security-by-design system testing with digital twins, some preliminary considerations are required. Security system testing presents a potential digital twin security use case [10], whereby security-by-design system testing can be located at early lifecycle phases (see Figure 1). Thus, the lifecycle accompanied by the digital twin is at an early phase (e.g., planning, design), where no physical counterpart yet exists[1].

This leaves the *simulation mode* the only option, as all other digital twin security modes (i.e., replication and analysis) rely on historic/state data produced from an existing counterpart [6]. Please note that although the simulation mode does not mirror the physical counterpart as it currently is in its state (as does the replication mode), it nevertheless allows for providing dynamic aspects and change of states. The latter can manifest, for example, in altering sensor values, network packages and log entries. Thus, as long as the simulation environment is able to present its (planned) counterpart accurately, the simulation mode is useful for system security testing and to detect potential future threats. This way, when implementing the real-world counterpart, it can be rendered secure-by-design.

With the help of the digital twin simulation of the planned counterpart, potential vulnerabilities can be revealed in advance when its security is tested. This nevertheless requires selecting simulation technologies to represent the planned counterpart that provide sufficient fidelity for meaningful security analyses. The choices made to build the digital twin that represents an ICS strongly affect the validity of the results obtained with the proposed approach. Thus, the selection should include considerations as to what has to be simulated (e.g., network, application and/or physical processing layer) with the level of granularity and which simulation technologies might fit to the given requirements. Of course, this is strongly dependent on the system to represent and its use case.

3.2 Proposed approach

Based on the preliminaries, Figure 2 illustrates our security-by-design testing approach with the digital twin. While the figure's left part presents the digital twin and security-by-design part (already explained in Section 3.1), the part on the right proposes the applied security testing method. At its core, we follow the four-stage penetration testing methodology, which represents the core of the NIST 800-115 guideline [28]. Thereto, techniques of overall security testing (examination, target identification, vulnerability analysis, security assessment) from the NIST 800-115 standard are added. The four main phases of the approach, including the appended techniques, are described in the following.

Planning. In this stage, rules, guidelines and overall testing goals are set to lay the foundations for successful security testing. An essential activity in this phase represents the planning of the later security assessment. The actual testing starts in the subsequent discovery phase.

Discovery. In the first part of this phase, information is collected. This involves scanning the network for active devices. There are

[1]This implies that a comparison between digital twin and physical twin is not possible yet.

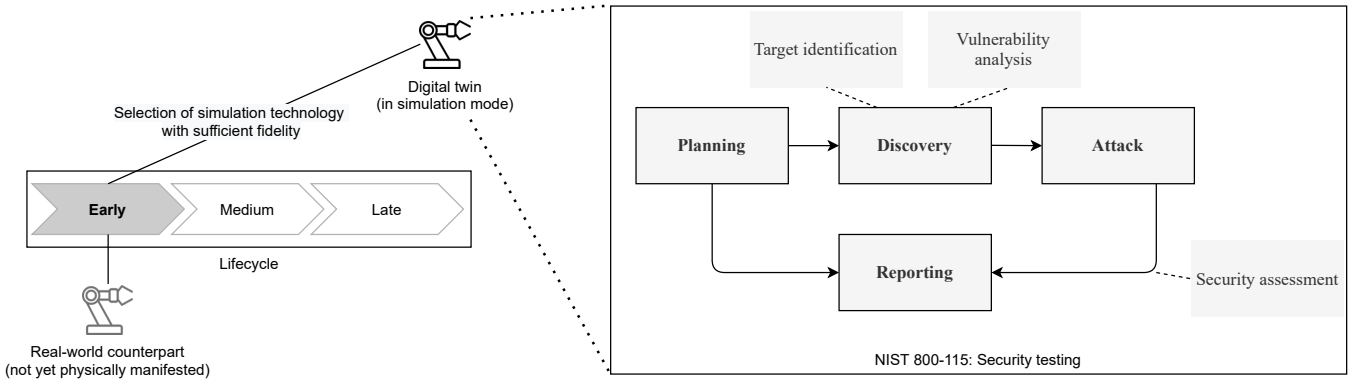

Figure 2: Proposed security-by-design testing approach with the digital twin.

passive and active approaches to discovering devices. Passive techniques use so-called sniffers that record network traffic and identify IP addresses, open ports and the possible relationships between devices. These passive techniques do not send their own information into the network and can read non-encrypted network traffic. Active techniques try to accelerate the collection of information, e.g. by sending normal, abnormal and illegal network traffic to a device in order to draw conclusions about the operating system from the response. Such scanning mechanisms enable *target identification*, where active devices, ports and services are identified. Depending on the network topology, a Man-In-The-Middle (MITM) position may be necessary to record network traffic.

In the second part of the discovery phase, a *vulnerability analysis* of the detected devices is performed. Public databases or special tools can be used to this purpose. Note that this does produce a list of potential, but not yet tested, system vulnerabilities.

Attack. This phase presents the core part of the methodology. It aims at the verification of the potential vulnerabilities identified during discovery. This attempted system exploitation can include activities such as forcing entry to the system, privilege escalation, browsing the system and installing tools to support security testing. These tools can serve to gather further access to systems or information. Next to proven vulnerabilities, their root causes should be determined.

All taken activities in this phase enable the *security assessment*. To enable structured assessment, the identified vulnerabilities and their causes can be categorized. For example, the categorization can orientate on the NIST SP 800-53 standard for security controls organization. Finally, the NIST National Vulnerability Database (NVD)[2] can support this process by providing information on Common Vulnerabilities and Exposures (CVE), and by scoring the vulnerabilities according to the Common Vulnerability Scoring System (CVSS).

Reporting. The reporting phase runs parallel to all three phases, and records the completed steps. It summarizes the potential and validated vulnerabilities and results in the final *security assessment*. While in the planning phase, the assessment is planned theoretically, the discovery and attack phases the conducted activities are recorded and reports are created regularly. After finalizing the attack phase, the overall report can be developed including the description of the identified vulnerabilities and the subsequent assessment of the risk the vulnerabilities present. Mechanisms and techniques for vulnerability mitigation conclude the report. The reported findings can be subsequently be used to improve the system's security.

[2]http://nvd.nist.gov/

4 IMPLEMENTATION

For demonstration purposes, a use case is provided in Section 4.1. To be able to show proof-of-concept, we developed a prototype based on the use case that can be used for security testing (Section 4.2).

4.1 Use case

Our digital twin represents an ICS setting of a pressure vessel, currently in the planning phase (early lifecycle), that is to be integrated to an existing industrial plant.

Figure 3 shows five levels concerning ICS and highlights the tackled levels of our digital twin. It relies on the general structure of ICS [18], which can be further divided into five layers [16]. We focus on levels 0, 1 and 2, which are intended to be integrated in an industrial plant that already provides levels 3 and 4. The levels of our pressure vessel setting (0-2) will further be focused on in the security testing (see Section 5).

At level 0, the pressure sensor (*Sensor*) and the safety valve (*Actuator*) are located. Both are monitored and controlled by a programmable logic controller (*PLC*) operating at level 1. If a certain pressure value in the vessel is exceeded (opening pressure), the PLC commands to open the safety valve at once to prevent any accidents. At level 2, the human machine interface (*HMI*) provides the current pressure value, the opening pressure and the status of the safety valve (closed/open) for human operators. For communication between the components, the network protocol of the existing plant (Modbus TCP/IP) is intended for usage.

4.2 Digital twin prototype

To prove our concept, we implemented a digital twin prototype of a planned pressure vessel. Therefore, we selected the `OpenPLC` technology amongst others. Below we explain each technology in more detail, and argue that they provide sufficient fidelity to

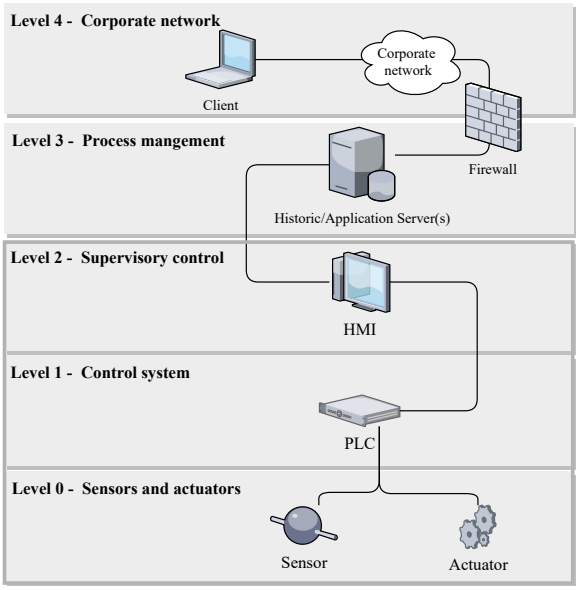

Figure 3: Focused ICS levels.

the potential real-world as required by a digital twin. In contrast to many other potential tools, the selected technologies remain close to the real world: the included ladder logic, network protocols, for instance, differ in no way to those in the real world. Thereby, `OpenPLC` is well used in academics, and its implementation as soft PLC (no embedded platform required) suits the digital twin purpose. To conclude, the obtained results can be relied upon to built a secure-by-design real-world ICS.

The topology and tools of our prototype of the pressure vessel is illustrated in Figure 4. We made our digital twin prototype publicly available at GitHub[3].

Utilizing these tools allows for proper integration of the Modbus TCP/IP network protocol and can therefore represent the network traffic realistically. Moreover, the logic of the PLC and the function of the HMI can be simulated close to real-world.

To represent the focused ICS levels (see Figure 3), each component (PLC, HMI and I/O simulator) runs on a virtual machine (VM), which operates in bridged network mode. The net filter driver manages the exchange of network packets and ensures that each VM is assigned its own IP-address on the network. This simulates that the host is physically wired to the network interface card. Communication between the individual components (VMs) is realized with Modbus TCP/IP. For simulating the PLC, the program `OpenPLC`[4] is used. The HMI is simulated via `ScadaBR`[5]. The I/O-simulator makes sure that the physical inputs and outputs are virtually represented.

Modbus TCP/IP. The request-and-response protocol relies on function codes and data fields [20]. Thereby, a function code can take any decimal value from 1 to 255 (1 byte). Modbus transactions are initiated by a clients request to the server. The function code contained in the request indicates the action to be carried out by the

[3]https://github.com/Leon1278/DigitalTwin-SecurityTesting
[4]https://www.openplcproject.com/runtime/
[5]https://sourceforge.net/projects/scadabr/

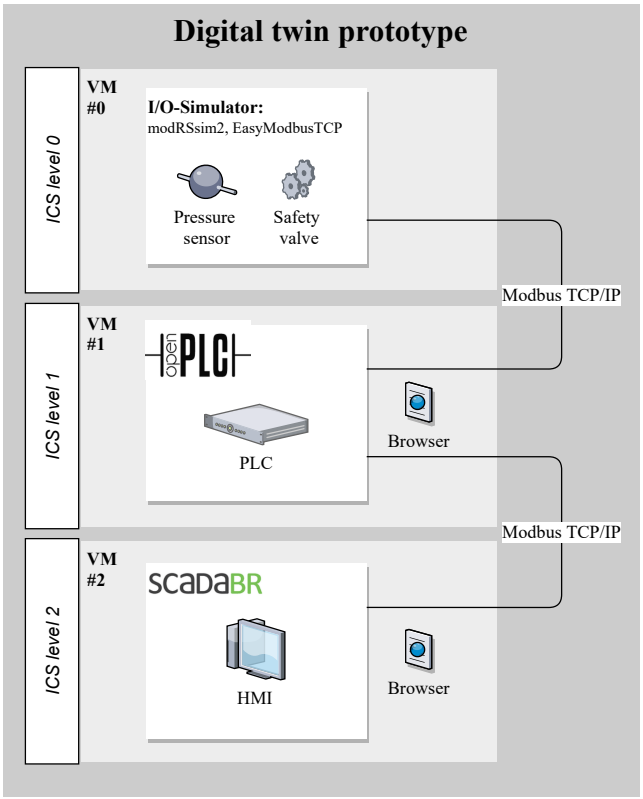

Figure 4: Topology of the digital twin prototype representing a pressure vessel.

server and the desired data. Function code values ranging from 1 to 65 are reserved for standard actions. Upon receiving the request, the server performs the action, responds with the same function code and the requested data. Modbus can be mapped via TCP/IP and is a prevalent industrial standard for ICS communication [25]. Network communication over port 502 indicates that Modbus is used[6]. By using the Modbus TCP/IP protocol, the network can be set up exactly as in the planned real world – so that the implemented digital twin environment produces the same network packages.

I/O simulator. The digital twin simulates the pressure gauge and the security valve, so that the physical process and the sensor values are virtually produced. To this end, we utilize `modRSsim2`[7] and the `EasyModbusTCP Server Simulator3`[8]. With these tools, the sensor data, which will appear in the PLC Modbus server can be simulated and directly tracked. This way, they support the development of systems using the Modbus protocol. While in real-world settings, sensors and actuators are directly bound to the PLC, it is not possible in simulations. However, this does not mean that the system cannot be properly represented as the I/O simulator can be virtually bound to the PLC by using Modbus. This is the only aspect that needs to be kept in mind for later security analyses, so that the Modbus

[6]The Internet Assigned Numbers Authority (IANA) has allocated port 502 for Modbus.
[7]https://sourceforge.net/projects/modrssim2/
[8]http://easymodbustcp.net/modbus-server-simulator

connection between PLC and I/O simulator is neglected as it will not exist in real-world.

OpenPLC. OpenPLC is used in many academic projects and meets the IEC standard 61131-3 for PLCs in industrial plants [1]. This ensures that it covers all relevant aspects of a PLC used in industry (e.g., ladder logic) and it provides sufficient fidelity for simulating a real-world PLC. The OpenPLC editor supports developing programs that can convert measured input to output. The developed program can be uploaded to the OpenPLC runtime to virtually operate a PLC. The OpenPLC runtime can even run on dedicated hardware (e.g. Arduino, RasperryPI) or – as in this digital twin prototype – virtually in a VM. Thus, the complete logic could be implemented in any real-world PLC without any adjustments.

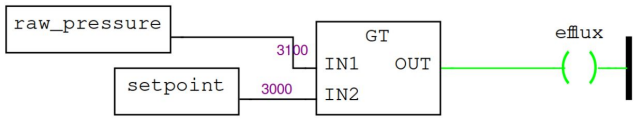

Figure 5: Ladder logic of the pressure vessel (program flow indicated in green).

The control logic of the PLC is created using ladder logic. Figure 5 shows the program with an open safety valve (efflux) at runtime. At core lies the Greater Then (GT) function block to compare two values: The measured pressure in the vessel (raw_pressure) and the specified opening pressure (setpoint). If the pressure exceeds the specified value, the GT block output is True, which opens the safety valve. As the current raw_pressure (3100) amounts greater than the setpoint (3000), the GT function block initiates the opening of the safety valve. The IP-address of the PLC in this setting is at 192.168.1.7.

ScadaBR. The prototype also provides a graphical interface within the HMI, which is realized using the open-source tool ScadaBR. Commonly, ScadaBR is used for simulating a Supervisory Control and Data Acquisition (SCADA) system, which possesses more features than a simple HMI. However, it can also be used to simulate a HMI, as in our case. Similarly to the argumentation for selecting OpenPLC, ScadaBR provides the same fidelity standards for simulation and is often used in academia. This is thanks to the fact that these to technologies are meant to be combined and featured in the OpenPLC-project[9].

Our HMI features a dashboard visualizing the state of the pressure vessel to the human operator (see Figure 6). The dashboard shows the current values of the measured pressure in the vessel, the opening pressure and whether the security valve is closed or open. The IP-address of the HMI in this setting is at 192.168.1.8.

5 PROOF-OF-CONCEPT

To prove our concept, we pass each step of our presented approach (see Figure 2). Thereto, we use the digital twin prototype and use case of the previous section. Thus, Section 5.1 demonstrates the planning phase. The subsequent Section 5.2 deals with the discovery

[9]https://www.openplcproject.com/

Figure 6: HMI dashboard visualizing the states of the pressure vessel.

phase, including target identification and vulnerability analysis. Section 5.3 provides the core phase of our framework, where we apply different attacks on our digital twin pressure vessel. Finally, a report assessing the security and proposing mitigation techniques is drawn in Section 5.4.

5.1 Planning phase

In this phase, we develop the rules and goals of our security test. We further divide these into attacker modeling and the targeted vulnerabilities. Technically, the attacker is implemented by another VM on the digital twin protoype and runs an OS specialized for security testing (Kali Linux[10]).

Attacker model. For the attacker model, we adopt the Dolev-Yao model [8], where the attacker possesses infinite computing capacity, time and insight into the attacked system, but cannot break the cryptographic methods used. The attacker on our ICS use case is defined as a group of human actors working together towards the goal of preventing the safety valve from opening – despite the measured pressure amounts higher than the opening pressure setpoint.

Thereby, we assume that the attacker has already been able to overcome levels 4 and 3 of the overall industrial plant (see Figure 3). As an outsider, the attacker subsequently tries to find out how levels 2, 1, and 0 of our ICS use case communicate with each other and how it can be attacked successfully.

Targeted vulnerabilities. In our setting, the attacker aims at exploiting the Modbus TCP/IP communication. In its current state of development, Modbus TCP/IP does not show any cryptographic security mechanisms [20].

5.2 Discovery phase

Target identification. As an initial step to collect information, a passive scanning approach with the network scanner nmap3[11] is used to check the local IP addresses. Potential running Modbus TCP/IP communication is indicated as port 502 is open on two hosts (PLC: 192.168.1.7, HMI: 192.168.1.8).

Vulnerability analysis. Using Wireshark[12], individual packets are intercepted and examined in more detail to look for potential vulnerability exploits. Figure 7 shows an Wireshark excerpt of a Modbus TCP/IP-Request from the IP 192.168.1.8 (HMI) towards

[10]https://www.kali.org/
[11]https://nmap.org/
[12]https://www.wireshark.org/

the destination IP 192.168.1.7 (PLC) with function code 04[13]. From regarding the address of the first register to be read (00) and the number of the register to be read (00), it follows that only the first value in the first input register is requested.

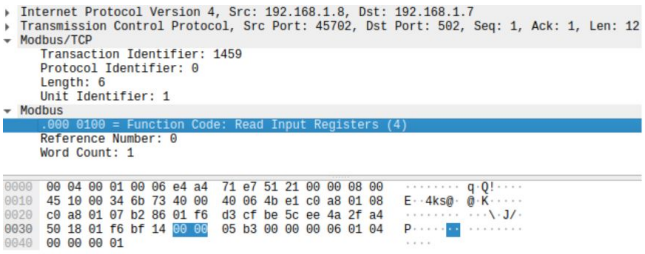

Figure 7: Modbus request sniffed using Wireshark.

The subsequent network packet (Figure 8) contains the corresponding response of the 192.168.1.7 (PLC) to IP 192.168.1.8 (HMI). It is replied with the unencrypted transmitted value 0x05 0xdc (1500). Thus, the attacker gathers the current pressure value of our vessel as it corresponds to the requested value in the register to be read. Wireshark is able to intercept this request-response scheme every 1000ms, reflecting the frequency of the query by the Modbus TCP/IP server. Using public sources, the attacker can find informa-

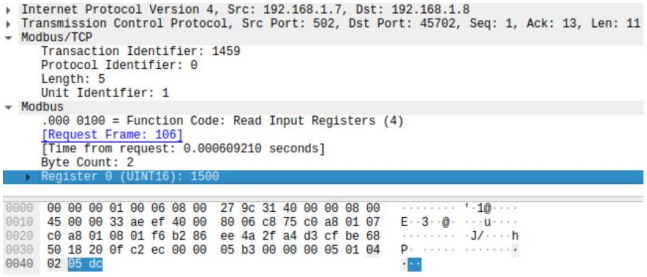

Figure 8: Modbus response sniffed using Wireshark.

tion on how to potentially exploit the vulnerabilities of Modbus TCP/IP to harm the system.

5.3 Attack phase

In this phase, the attacker makes use of the open-source security testing software `Metasploit`[14] to exploit the Modbus TCP/IP communication between the HMI and PLC. For the advanced attack, a custom C++ script[15] is applied that enables the constant altering of the safety valve.

Metasploit. In the world of penetration or system security testing, `Metasploit` represents a widely adopted framework, which is regularly kept up to date. The software provides exploits for various OS and applications. Several of those exploits can be successfully used to highlight the vulnerabilities of Modbus TCP/IP.

First, the unit identifiers (UID) of the Modbus-using devices (identified in the previous phase) can be determined in order to be able to carry out targeted attacks. Afterwards, the current pressure can be read from the input register using `READ_INPUT_REGISTERS`.

```
msf6 auxiliary(scanner/scada/modbusclient) > set action
     WRITE_REGISTER
action => WRITE_REGISTER
msf6 auxiliary(scanner/scada/modbusclient) > set DATA
     4000
DATA => 4000
msf6 auxiliary(scanner/scada/modbusclient) > exploit
[*] Running module against 192.168.1.7
[*] 192.168.1.7:502 - Sending WRITE REGISTER...
[+] 192.168.1.7:502 - Value 4000 successfully written at
     registry address 1
[*] Auxiliary module execution completed
```

Listing 1: Manipulation of the register value.

Beyond reading, tables can also be overwritten with new values. The default opening pressure (`setpoint`) for the pressure vessel safety valve can be manually changed to 4000 (see Listing 1). Listing 1 uses the commands with the `WRITE_REGISTER` module to overwrite the value of 3000. The HMI of our digital twin then displays this changed opening pressure of the vessel, but does not give any alerts.

Although the value of the opening pressure can be successfully manipulated, sensor values like the current pressure are of constantly gathered and updated by the PLC Modbus server. Thus, manipulations of non-permanent sensor values are quickly overwritten the with the valid sensors. To overwrite such values constantly in the course of an attack, repetitions in high frequency are required.

Custom C++ script. In addition to specialized software, scripts can be developed to read and change values in the tables of Modbus TCP/IP devices. This gives the attacker the possibility to manipulate or control the targeted device. With a customized C++ based script, we are able to carry out an advanced attack compared to those presented previously. Inspired by a script presenting a Modbus attack on a PLC[16], we adapted the attack to fit to the ICS environment of this paper.

The script exploits the missing authentication within the Modbus protocol between slave and master by transmitting Modbus packets to the master. It transmits packets that overwrite the register value of the safety valve (`efflux`) with 0. Consequently, the latter will not open even if the register value of the I/O simulator (the valid sensor values) exceed the opening pressure (`setpoint`). If the frequency of the sent packets is now adjusted so that more malicious than actual packets are sent, the proper working of the PLC is severely impaired. Listing 2 shows the `while` loop that enables the high-frequent transmitting of malicious packets. The frequency is regulated by a `sleep` statement. The packets are constantly sent until the script is stopped by the attacker.

[13]Function code 04 is used to read values from the input register table of the requested device.
[14]https://www.metasploit.com/
[15]https://github.com/Leon1278/DigitalTwin-SecurityTesting/blob/main/modbus_tcp_injection_attack.cpp
[16]https://github.com/thiagoralves

```
while(1){
  /******** SEND PACKET ********/
  data_len = sendto(socket_fd, packet1, sizeof(packet1),
       0, (struct sockaddr *)&server_addr, sizeof(
       server_addr));
  if (data_len < 0){
      printf("Error sending data on socket %d\n",
           socket_fd);
      perror("error: ");
  }else if (data_len > 0){
      printf("Sent: ");
      for (int i = 0; i < data_len; i++){
          printf("%02x ", packet1[i]);
      }
      printf("\n");
  }
  sleep_us(sleep_time);
}
```

Listing 2: Frequent transmits of the manipulated Modbus packet to the PLC (code fragment).

While the valid pressure values are correctly transmitted to the HMI during an attack, the safety valve remains closed. This is due to the fact that, although the PLC compares the actual pressure sensor value with the setpoint and tries to open the valve, the attacker script deliberately transmits the register value to close the valve at high frequency – so that the value to open the safety valve is overwritten at once. As a consequence, the safety valve does not open when the true threshold value is reached. Thus, the pressure vessel will yield and potentially explode.

5.4 Reporting phase

In the course of the attack phase of our security test, various vulnerabilites were identified regarding the planned use of Modbus TCP/IP. To sum up, a sabotage of the ICS can be achieved with moderate effort on the part of the attacker. In the documentation phase, these vulnerabilities are evaluated and potential protection mechanism are found.

Security assessment. For security risk assessment, we follow the risk levels of the German Federal Office for Information Security (BSI), where vulnerabilities are classified in regard to their potential for entry and their damaging potential. The three main vulnerabilities identified are further evaluated in the following and summarized in Table 1. These consist of:

(1) missing transport encryption of the packets
(2) missing authentication between the communication partners
(3) missing mechanism to prevent mass sending of packets with unlimited frequency

In terms of entrance potential, the aspects status and dissemination method of the vulnerability are focused. To exploit vulnerability *(1)*, attacker tools such as the `Wireshark` are openly available, which allow gathering information such as the UID or the function codes of unencrypted Modbus traffic. Nevertheless, the attacker has to carry out these tools manually and requires general knowledge of networks and the Modbus protocol – resulting in a medium entry potential. Automated scripts exist to exploit vulnerability *(2)* and *(3)*, so that the potential for entry is high.

Next, the damaging potential is assessed by regarding potential loss and context. Vulnerability *(1)* compromises confidentiality throughout the network and thus, provides high potential for damage. Vulnerability *(2)* entails a very high potential for damage as missing authentication renders it possible to inject false data and therefore, to take control of the PLC. Vulnerability *(3)* also results in very high damage potential. The possibility of sending any number of packets at unlimited frequency, which are processed by the targeted Modbus server, renders taking control of the network conceivable.

Table 1: Security risk assessment of three identified vulnerabilities.

Aspect	Vuln. *(1)*	Vuln. *(2)*	Vuln. *(3)*
Potential for entry	medium	high	high
Damaging potential	high	high	high
Overall assessment	high	very high	very high

Finally, the determined entry potential of a vulnerability is combined with its damage potential. This results in a current damage potential from which risk levels can be derived. Vulnerability *(1)* results in a high risk. Vulnerability *(2)* and *(3)* each entail a very high risk. In our ICS scenario, a further clientele-specific adaptation of the risk might be beneficial. For example, if levels 3 (process control) and 4 (corporate network) are challenging to overcome, it can reduce the risk of using Modbus TCP/IP.

Potential mitigation mechanisms. In the present scenario, changes to the planned ICS are urgently recommended. The security test on the digital twin showed that known vulnerabilities can be exploited by an attacker to overtake main functions, and even damage the ICS.

For risk reduction and mitigation, the Modbus security protocol [21] might be installed. This represents an alternative protocol secured by transport layer security (TLS), although not widely adopted. However, this requires that the existing industrial plant can deal with the protocol. It might further include migrating to the new protocol in the existing industrial plant components. Additionally, it must be ensured that performance is not hampered. Implementing the TLS-encrypted Modbus protocol inhibits reading clear sensor values. Nevertheless, the production of overhead might provide issues with the newly implemented protocol, so accessing the systems remotely should not be practiced.

6 DISCUSSION

Some aspects of our work require further discussion, including the security testing scenario of our proof-of-concept (Section 6.1), the applicability of our approach for systems in general (Section 6.2) and the usage of the digital twin (Section 6.3).

6.1 Threat landscape

The focused scenario for the security test remains a valid problem in practice. For instance, of all ICS accessible via Shodan[17], approx. 25% rely on the unencrypted Modbus TCP/IP protocol for communication. On the 2020 market share concerning industrial Ethernet protocols, Modbus TCP/IP takes up approx. 8% of worldwide use

[17] https://ics-radar.shodan.io/

[24]. Above, we have provided a use case and shown how the security test phases can be conducted with the help of a digital twin. Nevertheless, this comprises only a proportion of what is feasible during the attack phase. With successful exploits on the Modbus TCP/IP communication like presented in our proof-of-concept security testing, further attacks and even entry to the ICS deploying the Modbus server (our PLC) are possible. Moreover, the attacker might also focus on the HMI to transmit false sensor values by carrying out a man-in-the-middle (MITM) attack, and disguising with the IP of the PLC (e.g., ARP spoofing). Also, less advanced malicious activities like simple denial-of-service (DOS) attacks can be carried out. Due to the sheer boundless possibilities of exploiting the identified vulnerabilities, we have focused on data manipulation as it shows how stealthy and advanced the attacks can get. Further, it shows that not only the network is hampered with, but that the attacker even can gain access to the register values of the Modbus TCP/IP server and overwrite these.

6.2 Industrial focus

While proof-of-concept is shown in regard to ICS in this paper, the approach is not limited to industrial settings. Although digital twins are mainly applied in industry [23], they can represent other systems as well. Independent from which system the digital twin represents, the approach is generically constructed and follows the guidelines from NIST 800-115 [28], which apply for the security testing of all systems. Our work focuses on industry as it provides an area, where security is still to be sufficiently addressed – especially in Industry 4.0. Also, the digital twin mainly being associated to industrial use cases [23], it provides high benefits in this area. This nevertheless excludes its use for other areas (e.g., enterprise IT systems).

6.3 Usage of digital twins

In terms of proposing the digital twin for carrying out security-by-design testing, several aspects are indicative. First, the digital twin provides the necessary focus on its (planned) real-world counterpart. As it accompanies its counterpart's lifecycle, the digital twin is moreover not only suitable for security testing in general, but especially for arranging security-by-design. Additionally, if security-by-design is carried out before physical existence of the real-world counterpart, system security testing in medium to later lifecycle phases (with real-world configurations and transmitted values from the subsequent existing physical system) renders testing even more efficient: If the digital twin testing already exists in an early lifecylce phase, the later adjustment when the real-world system is in place will be less time-consuming and less costly. Compare, for example, the digital twin to testbeds. Instead of being mere virtual solutions, testbeds typically rely on hardware and software for security(-by-design) testing, which increases costs. Moreover, testbeds do not accompany the system's lifecycle. So, in later lifecycle phases, essential tests are rather done on the real system. This often implicates that the industrial operation has to be stopped, which further entails considerable costs. Furthermore, in contrast to other potential solutions for security testing (like testbeds), the digital twin provides plenty potential of supporting other security applications during the lifecycle of its system-counterpart (see Figure 1). Nevertheless, the security assessment conducted in this paper could rely on other technologies than digital twins. However, regarding the opportunities concerning the usage of digital twins stated above, we argue that the implemented environment for testing is sufficient to be called a digital twin at its beginning – when the real world counterpart is yet to be built.

7 CONCLUSION AND FUTURE WORK

In this work, we have shown how to conduct security-by-design system testing with a digital twin. Thereto, we classify the security system testing and security-by-design system testing into the lifecycle of the system represented by the digital twin (see Figure 1). From there on, we propose an approach for carrying out security-by-design system testing with the help of a digital twin (see Figure 2). The approach utilizes a digital twin in simulation mode, which represents a system that has yet to physically manifest and should be secure-by-design (early lifecycle). By using the digital twin, the approach suggests to conduct the phases suggested by the NIST 800-115 guideline [28].

To provide proof of our conceptual approach, we implement a digital twin representing an ICS in planning (pressure vessel). We apply this use case to show how each security testing step can be properly addressed. Our use case focuses on identifying Modbus TCP/IP vulnerabilities of the communication between the PLC and the HMI of our setting. In the scenario, the attacker is able to compromise the system by manipulating sensor values and causing the system to malfunction. Overall, we have verified that digital twins can provide valuable security analyses to detect vulnerabilities of future systems and render systems secure-by-design.

As the proposed approach is generally not limited to industrial systems alone, other systems (e.g., enterprise systems) might follow the very same procedure to render planned systems secure-by-design. Nevertheless, it requires the creation of a digital twin at first. While this might prove an initial effort, the twin can be used throughout the system's lifecycle for further security testing.

Future work might carry out additional security tests on various different use cases. Moreover, the proposed approach might be extended by including security testing in the medium lifecycle phases of a system. While this will not alter the phases of security testing, it might require the consideration of other security operation modes of the digital twin (e.g., replication). Furthermore, it could include commands towards the now manifested physical counterpart. In this regard, (security) patch testing with digital twins might also be an interesting area to investigate. Besides, security analytics services for the industry (e.g., [11]) might benefit from integrating digital twins security applications like our proposed security-by-design system testing (see Figure 1).

ACKNOWLEDGMENTS

This work is partly performed under the ZIM SISSeC project (https://www.mobilitylogistics.de/logistics/produktionslogistik/projekt-sissec), which is supported under contract by the German Federal Ministry for Economic Affairs and Energy (16KN085725).

REFERENCES

[1] T. Alves, R. Das, and T. Morris. 2018. Embedding Encryption and Machine Learning Intrusion Prevention Systems on Programmable Logic Controllers. *IEEE Embedded Systems Letters* 10, 3 (2018), 99–102.

[2] M. Atalay and P. Angin. 2020. A Digital Twins Approach to Smart Grid Security Testing and Standardization. In *2020 IEEE International Workshop on Metrology for Industry 4.0 IoT*. 435–440.

[3] Stefan Boschert, Christoph Heinrich, and Roland Rosen. 2018. Next Generation Digital Twin. In *Proceedings of the 12th International Symposium on Tools and Methods of Competitive Engineering (TMCE 2018)*. 209–217.

[4] W. Danilczyk, Y. Sun, and H. He. 2019. ANGEL: An Intelligent Digital Twin Framework for Microgrid Security. In *2019 North American Power Symposium (NAPS)*. 1–6.

[5] Marietheres Dietz and Günther Pernul. 2020. Digital Twin: Empowering Enterprises Towards a System-of-Systems Approach. *Business & Information Systems Engineering* 62, 2 (2020), 179–184.

[6] Marietheres Dietz and Günther Pernul. 2020. Unleashing the Digital Twin's Potential for ICS Security. *IEEE Security Privacy* 18, 4 (2020), 20–27.

[7] Marietheres Dietz, Manfred Vielberth, and Günther Pernul. 2020. Integrating Digital Twin Security Simulations in the Security Operations Center. In *Proceedings of the 15th International Conference on Availability, Reliability and Security* (Virtual Event, Ireland) *(ARES '20)*. ACM, New York, NY, USA, Article 18, 9 pages.

[8] D. Dolev and A. Yao. 1983. On the security of public key protocols. *IEEE Transactions on Information Theory* 29, 2 (1983), 198–208.

[9] Matthias Eckhart and Andreas Ekelhart. 2018. Towards Security-Aware Virtual Environments for Digital Twins. In *Proceedings of the 4th ACM Workshop on Cyber-Physical System Security (CPSS '18)*. 61–72. https://doi.org/10.1145/3198458.3198464

[10] Matthias Eckhart and Andreas Ekelhart. 2019. *Digital Twins for Cyber-Physical Systems Security: State of the Art and Outlook*. Springer International Publishing, Cham, 383–412.

[11] Philip Empl and Günther Pernul. 2021. A flexible Security Analytics Service for the Industrial IoT. https://doi.org/10.1145/3445969.3450427

[12] Daya Gupta, Kakali Chatterjee, and Shruti Jaiswal. 2013. A Framework for Security Testing. In *Computational Science and Its Applications – ICCSA 2013*, Beniamino Murgante, Sanjay Misra, Maurizio Carlini, Carmelo M. Torre, Hong-Quang Nguyen, David Taniar, Bernady O. Apduhan, and Osvaldo Gervasi (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 187–198.

[13] E. Kang, S. Adepu, D. Jackson, and A. P. Mathur. 2016. Model-Based Security Analysis of a Water Treatment System. In *2016 IEEE/ACM 2nd International Workshop on Software Engineering for Smart Cyber-Physical Systems (SEsCPS)*. 22–28.

[14] Maninder Jeet Kaur, Ved P. Mishra, and Piyush Maheshwari. 2020. The Convergence of Digital Twin, IoT, and Machine Learning: Transforming Data into Action. 3–17.

[15] Peter Kieseberg and Edgar Weippl. 2018. Security Challenges in Cyber-Physical Production Systems. In *Software Quality: Methods and Tools for Better Software and Systems*, Dietmar Winkler, Stefan Biffl, and Johannes Bergsmann (Eds.). Springer International Publishing, Cham, 3–16.

[16] V. S. Koganti, M. Ashrafuzzaman, A. A. Jillepalli, and F. T. Sheldon. 2017. A virtual testbed for security management of industrial control systems. In *2017 12th International Conference on Malicious and Unwanted Software (MALWARE)*.

[17] R. Langner. 2011. Stuxnet: Dissecting a Cyberwarfare Weapon. *IEEE Security Privacy* 9, 3 (2011), 49–51.

[18] S. McLaughlin, C. Konstantinou, X. Wang, L. Davi, A. Sadeghi, M. Maniatakos, and R. Karri. 2016. The Cybersecurity Landscape in Industrial Control Systems. *Proc. IEEE* 104, 5 (2016), 1039–1057.

[19] S. Miller, N. Brubaker, D. Kappelmann Zafra, and D. Caban. 2019. Triton actor TTP profile, custom attack tools, detections, and ATT&CK mapping. https://www.fireeye.com/blog/threat-research/2019/04/triton-actor-ttp-profile-custom-attack-tools-detections.html. Accessed: 2020-02-23.

[20] Modbus Organization. 2012. *MODBUS Application Protocol Specification V1.1b3*. https://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf.

[21] Modbus Organization. 2018. *MODBUS/TCP Security Protocol Specification V21*. https://modbus.org/docs/MB-TCP-Security-v21_2018-07-24.pdf.

[22] Andrés Felipe Murillo, Luis Francisco Cómbita, Andrea Calderón Gonzalez, Sandra Rueda, Alvaro A. Cardenas, and Nicanor Quijano. 2018. A Virtual Environment for Industrial Control Systems: A Nonlinear Use-Case in Attack Detection, Identification, and Response. In *Proceedings of the 4th Annual Industrial Control System Security Workshop* (San Juan, PR, USA) *(ICSS '18)*. ACM, New York, NY, USA, 25–32.

[23] Elisa Negri, Luca Fumagalli, and Marco Macchi. 2017. A Review of the Roles of Digital Twin in CPS-based Production Systems. *Procedia Manufacturing* 11 (2017), 939–948.

[24] Joakim Nideborn. 2020. Industrial network market shares 2020 according to HMS Networks. https://www.hms-networks.com/news-and-insights/news-from-hms/2020/05/29/industrial-network-market-shares-2020-according-to-hms-networks. [Online; accessed 11-Feb-2021].

[25] O. N. Nyasore, P. Zavarsky, B. Swar, R. Naiyeju, and S. Dabra. 2020. Deep Packet Inspection in Industrial Automation Control System to Mitigate Attacks Exploiting Modbus/TCP Vulnerabilities. In *2020 IEEE 6th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS)*. 241–245.

[26] Marco Rocchetto and Nils Ole Tippenhauer. 2017. Towards Formal Security Analysis of Industrial Control Systems. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security* (Abu Dhabi, United Arab Emirates) *(ASIA CCS '17)*. ACM, New York, NY, USA, 114–126.

[27] Juan E. Rubio, Rodrigo Roman, and Javier Lopez. 2018. Analysis of Cybersecurity Threats in Industry 4.0: The Case of Intrusion Detection. In *Critical Information Infrastructures Security*, Gregorio D'Agostino and Antonio Scala (Eds.). Springer International Publishing, Cham, 119–130.

[28] Karen Scarfone, Murugiah Souppaya, Amanda Cody, and Angela Orebaugh. 2008. Technical guide to information security testing and assessment. *NIST Special Publication* 800-115 (2008).

[29] Colin Tankard. 2015. The security issues of the Internet of Things. *Computer Fraud & Security* 2015, 9 (2015), 11 – 14.

[30] F. Tao, H. Zhang, A. Liu, and A. Y. C. Nee. 2019. Digital Twin in Industry: State-of-the-Art. *IEEE Transactions on Industrial Informatics* 15, 4 (2019), 2405–2415.

[31] Abebe Tesfahun and Lalitha Bhaskari. 2016. A SCADA testbed for investigating cyber security vulnerabilities in critical infrastructures. *Automatic Control and Computer Sciences* 50 (01 2016), 54–62.

[32] Thomas H.J. Uhlemann, Christian Lehmann, and Rolf Steinhilper. 2017. The Digital Twin: Realizing the Cyber-Physical Production System for Industry 4.0. In *Procedia CIRP*, Vol. 61. Elsevier B.V., 335–340.