2022 International Symposium on Power Electronics,
Electrical Drives, Automation and Motion

Enhancing Modbus/TCP-Based Industrial Automation and Control Systems Cybersecurity Using a Misuse-Based Intrusion Detection System

F. Katulić, D. Sumina, I. Erceg, S. Groš
University of Zagreb
Faculty of Electrical Engineering and Computing
Unska 3, 10000 Zagreb, Croatia.
Corresponding author: F. Katulić (filip.katulic@fer.hr)

2022 International Symposium on Power Electronics, Electrical Drives, Automation and Motion (SPEEDAM) | 978-1-6654-8459-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/SPEEDAM53979.2022.9842239

***Abstract*—Modbus over TCP (Modbus/TCP) is a very popular protocol in industrial automation and control systems (IACS), but at the same time it is completely unprotected in terms of cybersecurity. This allows adversaries to manipulate controlled processes by forging or modifying process values in the Modbus protocol data unit (PDU), potentially causing damage to IACSs. In this paper, we propose the use of a misuse-based intrusion detection system (IDS) to detect out-of-bound process values and in that way make it difficult for an adversary to manipulate process values. To test the feasibility of this approach, a cyber-physical system was created, simulating an IACS water treatment plant. The implemented rule-based alarms and warnings were based on the industrial process and an adversary threat model, focusing on the process values of the IACS. This approach shows a promise as an additional safety mechanism to standard IACS cybersecurity solutions.**

Keywords—Automation, Communication system security, Cyber-physical systems, Industrial communication.

I. Introduction

IACS are systems used to monitor and control specific technological processes in industry (e.g., manufacturing, transportation, energy distribution). With the development of IACS systems, proprietary industrial communication networks and protocols have been replaced by those that use Ethernet communication technology while interconnecting once isolated industrial communication networks with corporate information technology (IT) systems. One of the main problems with the transition from legacy industrial communication systems to newer Ethernet-based systems is that IACS were not designed with taking cybersecurity into account. In addition, industrial cybersecurity threats continue to evolve and the number of reported cybersecurity incidents is increasing [1]. Although there have been some initiatives to standardize industrial cybersecurity requirements, industrial cybersecurity came into spotlight with the Natanz nuclear power plant cyberattack, also known as the Stuxnet attack [2]. An unknown advanced persistent threat (APT) undermined Iran's nuclear development program by destroying one-fifth of the uranium enrichment centrifuges using a malware called Stuxnet. Next, the 2015 Ukraine Blackout attack targeted Ukrainian electric power distribution companies, affecting more than 200 thousand people by power outages [2]. One of the recent documented cybersecurity incidents is the Colonial Pipeline attack, which halted the pipeline's operations due to a ransomware attack on the IT equipment used to manage the pipeline [3].

This work has been fully supported by the European Regional Development Fund under the project "An Innovative Solution for Cyber Security Management of Industrial System of Facility and Process Automation" (KK.01.2.1.02.0009).

This paper gives an analysis of potential IACS cybersecurity enhancements based on the generation of rules for process values for the plant-specific misuse-based intrusion detection system (IDS). In contrast to previous studies in which the IDS was implemented at the IT level, this paper proposes an additional implementation of the IDS at the process level. A custom experimental setup was created, simulating an IACS of a water treatment plant. The potential security enhancements are tested and verified on the created cyber-physical system, enabling the creation of a novel methodology for IDS plant-specific deployment and rule generation. Finally, the results have shown significant capabilities of the implemented IDS, however, it needs to be stressed that the IDS does not mitigate the discovered vulnerabilities, but only allows the detection of occurring attacks.

The paper is organized as follows. In Section II, the related work is presented. Section III gives a cybersecurity analysis of the Modbus/TCP communication protocol. Section IV contains information about the selected Security Onion 2.3 IDS, with an analysis of the Modbus data ruleset. The security and performance analysis of the implemented cyber-physical system is presented in Section V. Finally, conclusion and future work are given in Section VI.

II. Related Work

To combat the advance of cybersecurity threats, researchers and industry are trying to find solutions to improve industrial cybersecurity.

Compared with IT systems, IACSs have a significantly longer lifetimes, even up to 30 years of continued service. For this reason, industrial networks need to be robust and have reduced maintenance downtime. These are the requirements that the Modbus/TCP industrial communication protocol satisfies [5]. On the other hand, the lack of integrity, authentication, and encryption of industrial protocols due to availability requirements makes Modbus/TCP particularly vulnerable to false data injection attacks (FDIA). In [5] the authors developed an attack model for the Modbus/TCP protocol based on the threat, attack, vulnerability, and impact, which confirmed the high vulnerability of the Modbus/TCP.

Next, in [6], the authors implemented a smart grid cyber-physical system based on the Modbus/TCP protocol and analyzed the system behaviour under selected cyberattacks. The authors emphasize the need for cyber-physical system testing, which may lead to the discovery of new unknown vulnerabilities. In addition to the IACS-specific cybersecurity solutions [7], IDSs could play an important role in protecting industrial communication networks.

In [8] the authors created a ruleset for a misuse-based IDS that covered most of possible attacks on the Modbus/TCP header but lacked any Modbus data field attack analysis. Further, in [9] the authors analyzed Modbus data field attacks, but the rule syntax was not adequate for direct editing by an IACS operations engineer. The operations engineer would need to manually convert each Modbus data value from decimal to hexadecimal value, and to calculate precise position of the Modbus data values within the TCP packet structure, which could lead to an error.

Furthermore, an anomaly-based IDS based on machine learning algorithms was developed in [2]. The authors analyzed network traffic when there was no attack. In the case of an attack, the trained model was able to recognize the attack and create an alert. The advantage of the anomaly-based IDS is that anomaly-based systems could in theory detect attacks that have not yet been seen and usually exploit zero-day vulnerabilities. The main disadvantage of anomaly-based IDSs is many false positive and negative alarms when compared to the misuse-based IDS [2].

III. Cybersecurity of the Modbus/TCP Industrial Communication Protocol

Modbus/TCP is a client/server Ethernet-based industrial communication protocol developed by a company named Modicon. The protocol was created by encapsulating a protocol data unit (PDU) of a proprietary Modbus-over-Serial-Line protocol using a TCP protocol [10]. The original Modbus-over-Serial-Line protocol had two different transmission modes: ASCII and RTU, of which RTU is still widely used when RS232/RS485 serial Fieldbus technology is implemented. Fig. 1 shows the structure of a Modbus/TCP packet. Modbus/TCP uses Ethernet technology (IEEE 802.3) for layers 1 and 2, Internet Protocol (IP) in layer 3, TCP protocol in layer 4, and Modbus/TCP application protocol in the upper layers of the OSI (Open Systems Interconnection) network model. The Modbus/TCP protocol itself has a special application header called the Modbus application header (MBAP), shown in Fig. 2. The function code and the associated Modbus data are encapsulated with the MBAP header to form the Modbus Application Data Unit (ADU).

Regarding the cybersecurity of communication between devices using the Modbus/TCP communication protocol, such communication inherits the cybersecurity vulnerabilities of the lower-level protocols and used technologies and introduces new cyber vulnerabilities inherent to the original Modbus protocol [7]. Based on the work in [7], [8] and the penetration testing of the Modbus/TCP protocol within the implemented cyber-physical system, several categories of attacks that an adversary threat could exploit have been classified. The attacker's position is assumed to be within the

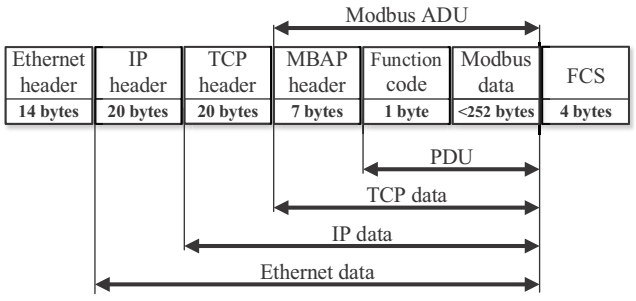


Fig. 1. Modbus/TCP packet structure

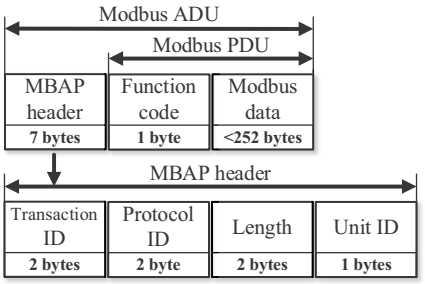


Fig. 2. Modbus/TCP application header format

industrial communication network (e.g. infected working station). The main reason for making this assumption are the results presented in [6] and [7] which show that the attacker's possibilities significantly rise after gaining access to communication networks. For example, initial access to communication networks connected to the Internet could be gained through phishing attacks. Still, offline networks could also potentially be affected by attacks generated by malware introduced through removable media [2]. The second assumption is that the tested experimental setup was created without taking measures to increase the created system's cybersecurity. Finally, the attacker does not know the network architecture of the attacked IACS.

Possible threat scenarios are defined using ISA/IEC 62443 [7] standard security levels (SL), which can qualitatively describe the required defence mechanisms within a zone when defending against threats with varying levels of knowledge, motivation, and financial resources. SLs are distributed from numbers 1 to 4, with SL 4 being required when defending against adversaries with large resources (such as government-funded APT groups). After conducting a high-level risk assessment of the created cyber-physical system, several threats have been identified. Table I shows three identified threats against which the Modbus/TCP communication protocol has no inherent protection mechanisms. Each of the three identified network-based threats could be further divided into different attack categories, and some attack techniques overlap, e.g., the Man in the Middle (MiTM) attack could also be used for Denial of Service (DoS) by dropping packets. DoS, spoofing, and information gathering attacks were carried out against the implemented cyber-physical system. The attack examples were chosen based on the attacker capability and the simplest way to maximize the caused damage to the IACS. The impact of the attacks was described using three different risk categories, in which the high risk indicates significant impact to the IACS based on the corporate risk matrices. Experiments were conducted using the open-source Kali Linux 2021.4a distribution with more than 400 built-in tools for penetration testing [11]. Based on the risk assessment, the attacker's capability is assumed to be SL-2.

TABLE I. IDENTIFIED THREATS

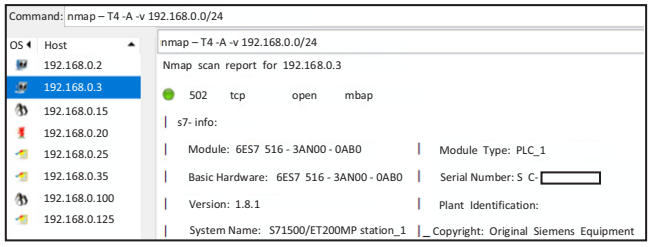
Threat	Attacker capability (SL)	Worst probable vulnerability source	Attack example	Risk
Denial of service	2	Lower-level protocols (TCP, Ethernet)	TCP SYN flooding	Medium
Spoofing	2	Lower-level protocols (TCP, ARP, Ethernet)	MiTM (FDIA)	High
Intel acquiring	2	Lower-level protocols (TCP ICMP, Ethernet)	Ping sweep, Wireshark	Low

The chosen DoS attack was the TCP SYN flooding, in which an attacker exhausts the server's resources by sending a large number of TCP packets coming from non-valid spoofed IP addresses with the flag SYN ON. The SYN flag indicates to the server that the client wants to initiate a connection. The server dedicates some resources for each connection, eventually depleting all the server resources. The attack exploits a lower-level TCP protocol vulnerability that was incorporated into the Modbus/TCP communication protocol. The attack was executed using the Hping 3 network testing tool built into Kali Linux.

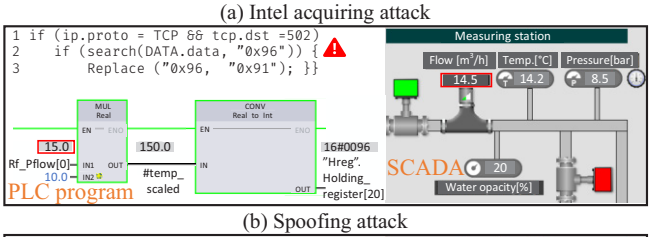
The second attack chosen was the MiTM spoofing attack. An adversary modifies the Modbus data field within the Modbus/TCP PDU, fooling the legitimate Modbus/TCP client and server. Based on the attacker's knowledge, one could replace the Modbus data field with random values, and the other could replace the values within the Modbus data field with legitimate process values (within the minimum and maximum value interval for a process value). It was assumed that the attacker does not know locations of the specific process values within the Modbus data field, nor the allowable intervals for process values. The attack was performed using a tool called Ettercap from an infected working station located within the network.

Finally, the third selected attack was the intel acquiring attack, in which an attacker used tools and scripts to discover and map the industrial control network devices within the plant. The results of the executed attacks are given in Fig. 3.

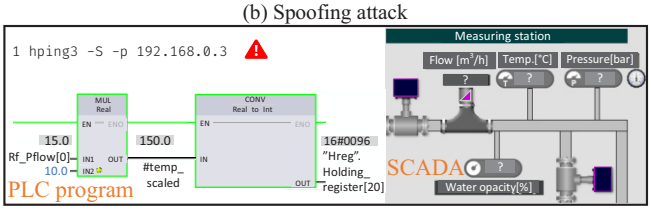
As can be seen in Fig. 3 (a), the attacker successfully obtained information concerning the industrial plant (IP and MAC addresses, firmware versions, network architecture, device roles, etc.). Afterwards, the attacker spoofed the communication between the Modbus/TCP client and the server, tricking the operations engineer located in the Supervisory Control and Data Acquisition (SCADA) room with different flow value (14.5) than the one in the plant (15). The SCADA interface, the spoofing command and the PLC program that enables the transmission of numbers rounded to the first decimal via Modbus/TCP are shown in Fig. 3 (b).



(a) Intel acquiring attack



(b) Spoofing attack



(c) Denial of Service attack

Fig. 3. Performed attacks

The last attack conducted was a DoS attack, in which the attacker flooded the Modbus/TCP server with a large number of TCP packets. As a result, the operations engineer lost track of the automated process, with the SCADA system becoming unresponsive. The DoS command and the interfaces are given in Fig. 3 (c). Several important conclusions can be drawn from these experiments. First, an industrial plant could become a victim of an adversary, whose capabilities, motives, and other resources correlate with the criticality of the industry type. Critical infrastructures, such as water treatment plants, power distribution networks, petrochemical plants, etc. are at a significantly higher risk when compared to other IACSs [3]. It is shown that one can use the free tools provided by Kali Linux OS to execute an attack that can disrupt communications when using the Modbus/TCP protocol, even with the knowledge that is not IACS-specific.

Second, the system itself does not provide signals about occurring attacks, at least not until it is too late to act upon them. Even if the system is configured according to the recommendations to increase its cybersecurity, it is critical to implement some form of an IDS system that can alert an operations engineer in the SCADA room. Once the alert is triggered, the operator needs to respond, keeping in mind that the severity of the potential cyberattack could have a much greater financial, environmental, safety, and reputational impact than stopping the plant production and further investigating the cybersecurity situation within the industrial plant. As can be seen, response time is critical when it comes to cybersecurity in IACSs, but it is also important that the operator is aware of potential IACS consequences depending on whether he/she acts or not.

IV. Intrusion Detection System

Network IDSs are used for information gathering, analysis, and automatic monitoring of communication data within the network. Usually, IDSs are divided into misuse-based and anomaly-based systems. Misuse-based IDSs detect some misuse of a system by user-defined set of rules. On the other hand, anomaly-based systems analyze and compare differences in the system's behaviour compared to the normal baseline state. They often use machine learning algorithms and other advanced data analysis methods. Network IDSs are usually implemented in standard IT networks and have shown significant capabilities when it comes to detecting an adversary within the network [2]. That being said, detection is the second step to mitigate a threat within the IACS, right after prevention. With this in mind, and considering the ease-of-use requirement (which is standard in IACSs), several free IDS tools that can be implemented in IACS networks that use Modbus/TCP for communication have been analyzed. There are some IACS-specific IDS solutions, but most of them are not free. The most widely used IDSs are Snort and Suricata, which are two independent open-source misuse-based IDS systems. Based on the analysis in [9] and the latency test performed (see Table II), significant differences between the two systems were not found.

TABLE II. LATENCY TESTS

Tool	Average detection time [ms]	Standard deviation [ms]	Maximum detection time [ms]
Snort	32.67	13.2	91.94
Suricata	81.28	19.12	121.8

The latency test was performed by sending one hundred standard ICMP packets (ping messages) to the chosen device within the network, and the average detection time was measured. The ICMP packet detection rule had the following format:

alert ICMP any any → any any (msg:"ICMP packet detection";) (1)

which was identical for both the Snort and Suricata IDS. The average detection time for Suricata IDS was higher than the time needed with the Snort IDS. Still, it was within the expected time interval in which an operations engineer could act upon (which is at least several seconds).

Based on the following and the fact that the Suricata IDS has a more advanced module for the Modbus/TCP data field analysis, we chose Suricata for the IDS within the system. Finally, the Security Onion 2.3 open Linux distribution, a free out-of-the-box IDS solution [9], was chosen. Security Onion 2.3 includes several tools for log acquiring, network monitoring and threat dealing, such as CyberChef, Zeek (formerly Bro), Wireshark, Suricata and Wazuh. Security Onion 2.3 was installed on the workstation located within the SCADA room using VMware tool.

Suricata's Modbus/TCP module can parse the entire Modbus PDU, allowing the creation of many different process-specific rules. Most papers [6], [8] dealing with the implementation of a misuse-based IDS in a Modbus/TCP-based IACS typically analyze the rules that defend against lower-level protocol attacks (attacks that target protocols such as TCP, IP, ARP, etc.) and attacks targeting the MBAP application header of the standard Modbus/TCP packet.

This study focuses on the Modbus data field, which is plant-specific. The Modbus data field transfers process values (16-bit words or 1-bit Boolean values) between Modbus/TCP clients and servers, whose interval, value size and position within the Modbus/TCP data field are specified by the automation program developer. Based on the automated process, it is possible to limit the maximum and minimum value of the process values exchanged between Modbus/TCP devices in the network, but only if the person creating the IDS rules knows the order in which the process values are sent. With that in mind, the IDS Modbus/TCP rules could have the following format:

alert MODBUS any any → $Modbus_Server 502 (msg:"The maximal value (50) for the FIC 0 water pump reference flow is violated"; modbus.access: write holding, address 30,value >50; sid:52; rev:1;) (2)

The alert keyword indicates that the rule is an alert and defines what will happen when the signature matches (e.g., rules other than alerts can be created for discarding packets in intrusion prevention systems). The MODBUS keyword indicates which communication protocol is used. The third part of the rule specifies the source/destination addresses and ports of the traffic. The message part is the text that will be displayed to the operator after the alert is triggered. The most important part of the Modbus/TCP rule is the access setting, which allows the operator to create the alerts based on the Modbus/TCP function code, the data type (one bit or 16 bit registers), the address within the Modbus data field, and finally, the value of the process values within the data field.

The specified rule will generate an alert message if the value interval for the process value (e.g., water pump setpoint) is violated. The rule format is relatively simple and allows the operations engineer to modify and change process value addresses and intervals based on the industrial process. In addition, the simplicity of the changing values is critical due to the short time intervals in which the IACS can be maintained (IACS are only stopped for maintenance a few times a year).

The problem with this solution is that it is not possible to specify Boolean values since the rule format only allows intervals to be defined. As a matter of fact, this method does not provide any mechanism for detecting the change of the Boolean value with the Suricata MODBUS keyword module, which is just as important as the register data change. It should be noted that the method described could be theoretically vulnerable to random FDIA MiTM attacks, which are analyzed in Section V.

V. Security and Performance Analysis of the Implemented Experimental Setup

The cyber-physical experimental setup is implemented as an automated water treatment plant to provide an experimental attack environment. Fig. 4 shows the block diagram of the IACS.

The automation and simulation of the selected industrial process were implemented using two Siemens S7-1513-1 PN programmable logic controllers (PLCs). Some elements of the water treatment plant were simulated as black-box systems (e.g. water filtration) while the pumping, measuring and dosing stations were created as fully functional elements. The first PLC (called the Simulation PLC) was used to simulate the water treatment process while the second PLC (called the Control PLC) was used for the process control. The programming solutions were created using the TIA Portal v16 tool. Both PLCs were located in the plant area, which is physically separated from the SCADA room. In addition to the two used PLCs, a PcVue SCADA system was implemented within the SCADA room for monitoring purposes.

The operations engineer can choose between two different modes of operation:

1) **Automatic mode**, in which the Control PLC automatically controls the process, while allowing the operator to manually change the water opacity level and the reference water flow.

2) **Manual mode**, in which the operator can change all the values that can be modified in the automatic mode and open or close the flow and relief valves.

Detailed process and instrumentation diagram (P&ID) of the implemented system is shown in Fig. 5.

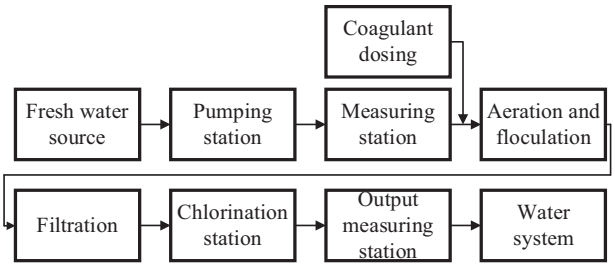


Fig. 4. The block diagram of the implemented IACS

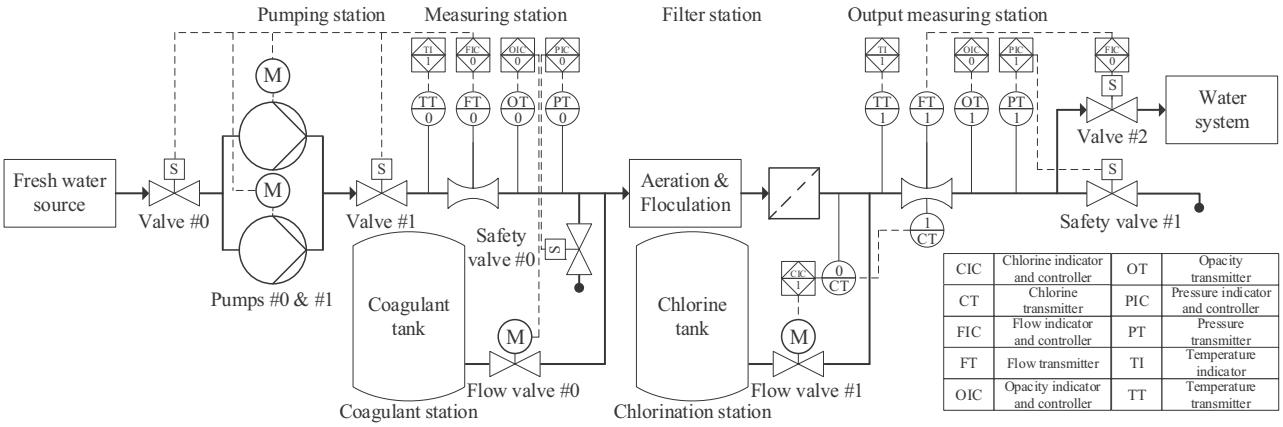


Fig. 5. The P&ID diagram of the implemented IACS

When talking about Modbus/TCP device application relations, it should be noted that:

1) **The Simulation PLC** is a Modbus/TCP server to the Control PLC.

2) **The Control PLC** is a Modbus/TCP client to the Control PLC and a Modbus/TCP server to the PcVue SCADA system.

3) **The PcVue SCADA system** is a Modbus/TCP client to the Control PLC.

Fig. 6 shows the network diagram of the implemented cyber-physical system. Port and traffic mirroring for the Security Onion IDS were established using the TP-Link T2500G-TS managed switch. Finally, all the unused Ethernet ports were disabled, based on the standard cybersecurity good practices given in [7].

After the experimental cyber-physical system was designed and implemented, threat modelling was performed based on Table I given in Section III. The primary threats were identified as potential DoS, spoofing and intel acquiring attacks. The potential capabilities of the attacker were modelled as SL-2 (medium resources). The attacks were performed under two different system conditions.

The results of the attacks on the IACS without an IDS, already presented in Section III, have shown that the intel acquiring and the spoofing attack were performed undetected. The DoS attack manifested itself in the way that the operations engineer in the SCADA room saw that the SCADA system was unresponsive.

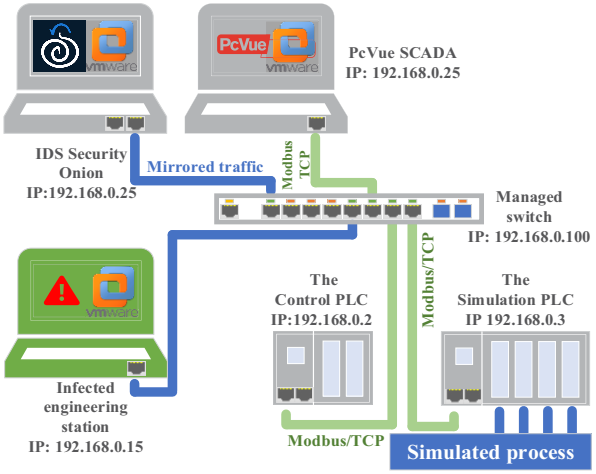


Fig. 6. Network diagram of the implemented cyber-physical system

Still, the unresponsiveness does not automatically mean that the system was subject to a cyberattack (it could be a communications failure). Furthermore, spoofing MiTM attack could be theoretically detected if the attacker changes the process value to a value that could not be achieved within the industrial process (e.g., the water temperature is set to 1000 degrees Celsius), but the operations engineer would have to notice the change in value, which would probably take time. Based on the given results, specific Suricata IDS rules were created, such as the rule (2) and the following rule:

alert tcp any any → $Home_NET 502 (flags: S; msg:"Possible DoS on IACS devices"; flow: stateless; detection_filter: track by_dst, count 50, seconds 1;) (3)

with (2) being the rule for the violation of the maximum flow reference value an MiTM spoofing attack could cause, and (3) being the rule for the detection of the DoS attack example. Rule (3) detects a possible DoS attack on the Modbus/TCP devices if the number of TCP packets with the flag SYN ON is higher than 50 within the time interval of one second. Process value rules were created for all the process values exchanged within the system, with the overall number of rules halting at number 82. The detection rules for the Nmap scanning methods have been analyzed by many authors [12], and due to the sensitivity of such subjects, they are not given. It should be noted that the Nmap scanning detection rules focus on the standard enumeration techniques (e.g. TCP SYN, NULL, FIN and Xmas scans). In this study, all the attack examples were executed fifty times, and the alerts were generated each time, except for the MiTM spoofing attack, which showed a 96% detection rate, as given in Table III. As expected, none of the attacks was mitigated, it only allowed the occurring attacks to be detected.

TABLE III. DETECTION RESULTS

Attack example	Custom IDS rule	Non-IDS detection	IDS detection	IDS detection rate	Mitigation
DoS – TCP SYN flood	Yes	Partly - SCADA system is unresponsive	Yes	100%	No
Spoofing - MiTM	Yes	Partly- user can detect off-value	Yes (partly)	96%	No
Intel acquiring -Nmap	Yes	No	Yes	100%	No

A DoS SYN flood attack was performed on the Control PLC, disrupting the communication with the PcVue SCADA system. Interestingly, the communication with the Simulation PLC remained unaffected. The reason for this is that the attack was aimed at port 502 of the Control PLC (server port to the client PcVue SCADA system), but the alternative port 2001 was chosen for the client-server communication to the Simulation PLC. Depending on the volumetric amount of the packets sent to the Modbus/TCP server and its intensity, the attack could be detected directly on the SCADA system, as shown in Fig. 3. Based on the alert generated within the IDS solution, the operator can verify that the system is under attack and proceed with the appropriate procedure if an attack occurs. With the SCADA system becoming unresponsive, the operations engineer would probably think of a communication failure and not of a cyberattack.

An MiTM spoofing attack was performed on the SCADA – Control PLC client-server communication path, changing the process values randomly. The first step for the rule generation was the definition of the minimum and maximum allowable values for a specific process value. The allowable interval was defined as an interval in which the values (e.g., flow, temperature, opacity) had a physical meaning and could assume a value within a defined interval. After defining the value intervals, rules such as (2) were implemented in the Security Onion Suricata IDS. The detection rate of 96% indicates high performance but shows that certain precautions should be taken. The proposed method works only if the attacker does not know the specific process values within the Modbus data field or the allowed intervals for specific process values. If the attacker changes process values within the specified limits, the FDIA attack will remain undetected. Depending on the automation solution, the attacker can damage the IACS by changing a process value within the allowed interval. If the dynamics of the automated industrial process are relatively slow, sudden changes to actuator signals could cause harm to the process and employees. Furthermore, the undetected attacks resulted from the attacker guessing a value that was within the defined allowed interval. The method itself does not protect the IACS from traffic-replay attacks either. Finally, it is not possible to detect changes to the Boolean values that are sent through the communication networks because the process values of 0 or 1 cannot be specified by the ruleset.

An intel acquiring attack was performed on the entire cyber-physical system, allowing for the discovery of important information about the IACS (e.g., PLC firmware versions, device application relationships, etc.). The intel acquiring attack was undetectable without the IDS solution, allowing the attacker to gain information that would later be used for other attacks. Although the IDS solution detected the Nmap-based intel acquiring attack, it should be noted that there are numerous tutorials and methods for IDS detection evasion, especially when using the Nmap tool [12].

The given results suggest that the IDS solution could improve cybersecurity within an IACS, especially in terms of monitoring and log management. Logs play a crucial role in finding a source of the attack in the case of a cyberattack. Secondly, the idea of a system that indicates to the operations engineer in the SCADA room that the IACS may be subject to a cyberattack is not as far as one could think of, which has been shown in this work.

VI. Conclusion and Future Work

In this paper, an analysis of potential Modbus/TCP security improvements based on a misuse-based IDS is given. The novelty of this paper is that the IDS is implemented at the process level, with the IDS rules created specifically for an industrial process. Potential security enhancements were tested and verified on a created water treatment cyber-physical system. The detection method for the DoS attack example was 100% accurate, for the MiTM spoofing attack 96%, and for the intel acquiring attack the detection method was 100% accurate. The high levels of detection rates indicate that the methods could potentially increase the cybersecurity of IACSs with a minimum impact on the availability requirement. Still, none of the methods mitigate vulnerabilities that were shown in the attack examples. Furthermore, the implemented IDS solution does not protect the IACS from traffic-replay attacks either. The misuse-based IDS cybersecurity solution should be used as an additional layer of security, with the help of standard IT solutions that do not interfere with the availability requirement within IACSs. Ideally, specific Modbus/TCP solutions for the integrity and authentication of the devices should be developed, with as low impact on the availability as possible.

Future work will be focused on the engineering of solutions that, in combination with more advanced rule-based IDSs, could protect IACSs from higher-level threats.

Acknowledgment

The authors would like to thank the companies S.C.A.N. d.o.o. and Autegra d.o.o. for their suggestions and support.

References

[1] S. McLaughlin *et al*., "The cybersecurity landscape in industrial control systems," in *Proc. of the IEEE*, vol. 104, no. 5, pp. 1039-1057, May 2016.

[2] F. Zhang, H. A. D. E. Kodituwakku, J. W. Hines and J. Coble, "Multilayer data-driven cyber-attack detection system for industrial control systems based on network, System, and Process Data," in *IEEE Trans. on Ind. Inform*, vol. 15, no. 7, pp. 4362-4369, July 2019.

[3] K. McGladrey, *Critical Infrastructure Requires Modernization*, IEEE, Sept. 27. 2021. Accessed on: Mar. 2022. [Online], Available: https://transmitter.ieee.org/critical-infrastructure-requires-modernization/

[4] M. Cheminod, L. Durante, L. Seno and A. Valenzano, "Performance evaluation and modeling of an ind. application-layer firewall," in *IEEE Trans. on Ind. Inform*, vol. 14, no. 5, pp. 2159-2170, May 2018,

[5] Huitsing, P.; Chandia, R.; Papa, M.; Shenoi, S. "Attack taxonomies for the Modbus protocols". *Int. J. of Crit. Infra. Protect.* 2008, 1, 37–44.

[6] B. Chen, N. Pattanaik, A. Goulart, K. L. Butler-purry and D. Kundur, "Implementing attacks for Modbus/tcp protocol in a real-time cyber physical system test bed," *2015 IEEE Int. Workshop Technical Committee on Communications Quality and Reliability (CQR)*, 2015.

[7] R. S. H. Piggin, "Development of ind. cybersecurity standards: IEC 62443 for scada and ind. control system security," *IET Conf. on Ctrl and Autom.. 2013: Uniting Problems and Solutions*, 2013, pp. 1-6

[8] T. H. Morris, B. A. Jones, R. B. Vaughn and Y. S. Dandass, "Deterministic intrusion detection rules for Modbus protocols,"*46th Hawaii Int. Conf. on Sys. Sci*, 2013, pp. 1773-1781.

[9] G. Ravikumar, A. Singh, J. R. Babu, A. Moataz A and M. Govindarasu, "D-ids for cyber-physical der Modbus system - architecture, modeling, testbed-based evaluation,"*Resill. Week*, 2020, pp. 153-159.

[10] The Modbus Organisation, *Modbus messaging on TCP/IP implementation guide V1.0b*, Oct. 24. 2006. Accessed on: Jan. 2022. [Online], Available:MODBUS Messaging Implementation Guide 1 0 b

[11] Offensive Security, Accessed on: Mar. 2022. [Online], Available: https://www.kali.org/docs/introduction/

[12] G. Lyon, *Subverting IDS*. 1. 2009. Accessed on: Mar 2022. [Online], Available: https://nmap.org/book/subvert-ids.html