electronics

MDPI

Article

Industrial Control Systems Security Validation Based on MITRE Adversarial Tactics, Techniques, and Common Knowledge Framework

Divine S. Afenu, Mohammed Asiri and Neetesh Saxena *

School of Computer Science and Informatics, Cardiff University, Cardiff CF24 4AG, UK; afenud@cardiff.ac.uk (D.S.A.); asirima@cardiff.ac.uk (M.A.)
* Correspondence: saxenan4@cardiff.ac.uk

Abstract: Industrial Control Systems (ICSs) have become the cornerstone of critical sectors like energy, transportation, and manufacturing. However, the burgeoning interconnectivity of ICSs has also introduced heightened risks from cyber threats. The urgency for robust ICS security validation has never been more pronounced. This paper provides an in-depth exploration of using the MITRE ATT&CK (Adversarial Tactics, Techniques, and Common Knowledge) framework to validate ICS security. Although originally conceived for enterprise Information Technology (IT), the MITRE ATT&CK framework's adaptability makes it uniquely suited to address ICS-specific security challenges, offering a methodological approach to identifying vulnerabilities and bolstering defence mechanisms. By zeroing in on two pivotal attack scenarios within ICSs and harnessing a suite of security tools, this research identifies potential weak points and proposes solutions to rectify them. Delving into Indicators of Compromise (IOCs), investigating suitable tools, and capturing indicators, this study serves as a critical resource for organisations aiming to fortify their ICS security. Through this lens, we offer tangible recommendations and insights, pushing the envelope in the domain of ICS security validation.

Keywords: Industrial Control Systems (ICSs); Indicators of Compromise (IOC); Denial of Service (DoS); ARP poisoning; Modbus; security validation; MITRE ATT&CK

Citation: Afenu, D.S.; Asiri, M.; Saxena, N. Industrial Control Systems Security Validation Based on MITRE Adversarial Tactics, Techniques, and Common Knowledge Framework. *Electronics* **2024**, *13*, 917. https://doi.org/10.3390/electronics13050917

Academic Editors: Juan-Carlos Cano and Dimitra I. Kaklamani

Received: 15 January 2024
Revised: 15 February 2024
Accepted: 26 February 2024
Published: 28 February 2024

1. Introduction

Industrial Control Systems (ICSs) are critical infrastructure components used in sectors such as oil and gas, energy, transportation, and manufacturing. As these systems become increasingly interconnected, the risk of cyber threats targeting ICSs has significantly grown [1]. In recent years, sectors relying on ICS have faced several forms of attacks, which include an attack on Rosneft Deutschland in mid-March 2022 [2] and an attack on the railway infrastructure of Belarus [3]. It is therefore imperative to validate the security of ICSs to ensure their resilience and protect critical operations. Focusing on this, we modelled two major attack scenarios on ICS and utilised various security tools to validate defences against these attack scenarios.

The MITRE ATT&CK framework, originally developed for enterprise IT environments, has been extended to address ICS security challenges. It provides a standardised methodology and knowledge base that allows for a structured assessment of ICS security controls, detection capabilities, and response procedures. This paper aims to leverage this framework to model and validate Denial of Service (DoS) and address resolution protocol (ARP) poisoning cyber-attack scenarios and explore Indicators of Compromise (IOCs) in ICSs using open-source detection tools. This paper also fills a gap in the academic discourse and offers practical recommendations, thus enhancing the resilience of ICSs against the evolving landscape of cyber threats [4].

This paper is divided into six sections. Section 1 introduces the paper with a brief description of ICS security validation based on the MITRE ATT&CK framework. Section 2 discusses fundamental terms in ICSs and reviews related work on ICS security and attack validation. Section 3 describes the system model and attack model, focusing on the DoS attack and ARP poisoning attack scenarios. Section 4 elaborates on the experimental design and the execution of the attacks. Section 5 details the execution and validation of the attacks. Finally, Section 6 concludes the paper.

1.1. Motivation

The motivation for this study is anchored in the escalating threats in the realm of ICSs, where the increasing sophistication of cyber-attacks and the unique vulnerabilities of these systems necessitate a more focused and specialised approach to security validation. The MITRE ATT&CK Framework, adapted to address the specific challenges faced by ICS, has offered a structured approach to ICS security, yet its practical application in real-world settings remains underexplored and holds significant potential for advancing ICS security practices [5]. A key driver for this work is the gap in the current literature regarding the practical application of security validation methodologies in ICSs. While theoretical models and simulations are prevalent, empirical studies that test these models against real-world attack scenarios are lacking.

This paper aims to bridge this gap by exploring both theoretical and practical aspects of ICS security validation, thereby providing a more comprehensive understanding of the vulnerabilities and defences in ICS environments [6]. Furthermore, this study is motivated by the need to delve into the understanding of IOCs specific to ICS environments. The existing research often generalises IOCs across different IT and ICS environments, overlooking the unique characteristics and vulnerabilities inherent to ICSs [7]. By focusing on ICS-specific IOCs and employing open-source detection tools, this paper seeks to offer nuanced insights into these systems' vulnerabilities.

In summary, the motivation for this paper is twofold: firstly, to address the pressing need for practical security validation in the face of evolving cyber threats targeting ICSs; and secondly, to deepen the understanding of ICS-specific security challenges and contribute to the development of effective security strategies.

1.2. Problem Statement

Despite the advancement of security in the ICS domain, systems have become increasingly interconnected, IT systems are also being integrated with Operation Technology (OT) systems, the environments have become more complex, and the risk of cyber threats has significantly grown. A single effective breach in this environment could result in severe repercussions, disrupt vital services, inflict economic harm, and potentially cause physical harm to individuals or equipment. However, with the implementation of effective security, IOC monitoring, and validation, these systems can be controlled effectively.

Some open-source tools, such as Suricata, Snort, Wazuh, and Wireshark, provide security monitoring and validation of DoS attacks and ARP poisoning Man-in-the-Middle (MITM) attacks in the ICSs environment. In this paper, the main problem that is addressed is validating IOCs in an ICS attack scenario using open-source tools.

1.3. Our Contribution

In response to the escalating challenges posed by the increasing interconnectivity of ICS and the integration of IT with OT, this paper addresses the importance of security validation within this environment. A successful breach in this intricate ecosystem could lead to severe repercussions, including the disruption of vital services. In this context, our paper focuses on the implementation of effective security measures, IOC monitoring, and validation to enhance the resilience of ICS environments. We recognise that the integration of IT systems with OT systems has introduced new vulnerabilities, making it imperative to devise robust strategies for safeguarding against cyber threats. While open-source tools

such as Suricata, Snort, Wazuh, and Wireshark have proven effective in monitoring and validating DoS attacks and ARP poisoning attacks in ICSs, our paper addresses the crucial importance of validating IOCs in the event of an ICSs attack scenario. By focusing on IOCs, we aim to contribute insights that:

- Empower organisations to effectively detect and respond to potential security incidents in their ICS infrastructure.
- Emphasise the importance of leveraging open-source tools for IOC validation, thereby enabling organisations to strengthen their defence mechanisms against evolving cyber threats in ICSs at a reduced cost.
- Through practical implementations and empirical evaluations, provide valuable contributions to the field, helping to secure critical infrastructure and mitigate the potential consequences of cyber-attacks in the ICS domain.

2. Background and Related Work

This section presents the fundamental terms used in ICS along with associated research conducted in this domain.

2.1. Related Terminology

ICS. ICSs play a crucial role in the functioning of critical infrastructures, making them prime targets for cyber threats. These are responsible for monitoring and controlling complex physical processes within industrial environments, enabling efficient and reliable operations, and ensuring the safety of personnel and the environment. They encompass a variety of components, including sensors, actuators, Supervisory Control and Data Acquisition (SCADA) systems, Programmable Logic Controllers (PLCs), Human–Machine Interfaces (HMIs), etc., as shown in Figure 1. These components work together to facilitate the seamless control and monitoring of industrial processes.

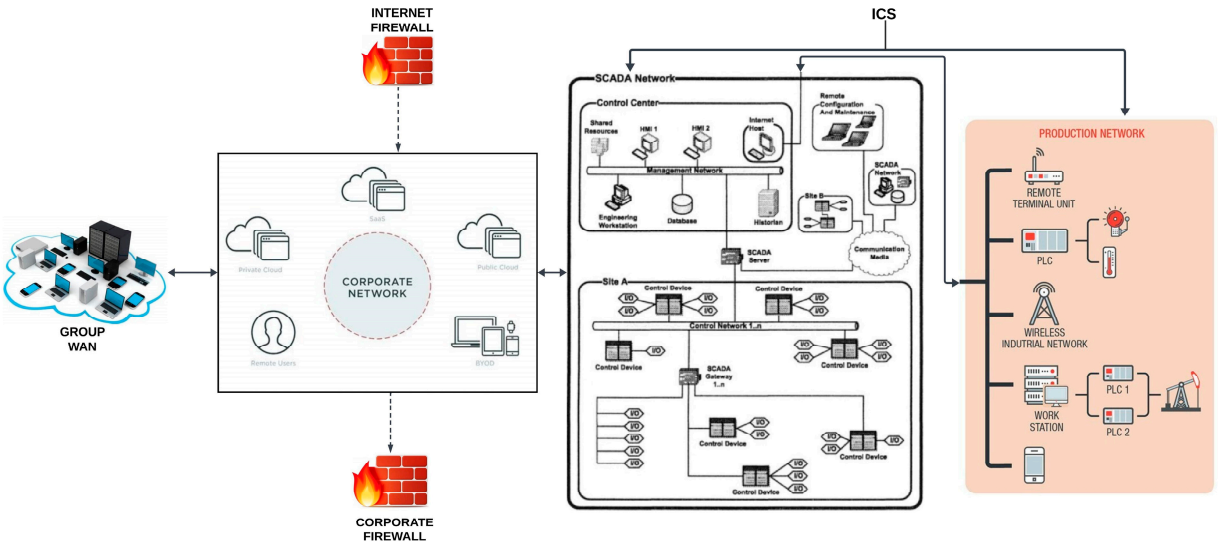


Figure 1. ICS architecture.

SCADA. SCADA systems have emerged as central players in modern industrial automation, especially in sectors like power generation and oil production [8,9]. They offer a holistic view of vast industrial setups by collecting data from devices like sensors and remote terminal units (RTUs) [10]. These data are processed in real-time, empowering operators with timely insights into decision-making [10]. A hallmark of SCADA is its data acquisition capability, which offers instantaneous insights while also chronicling historical data for trend discernment [11]. With built-in alarm management, SCADA ensures rapid alerts during anomalies, reducing potential operational hazards [12]. However, with the increase in Internet of Things (IoT) devices, SCADA systems face heightened cybersecurity

threats [13]. While modern SCADA architectures incorporate advanced security, legacy systems might be exposed to vulnerabilities [13]. As the industrial landscape evolves towards Industry 4.0, the integration of SCADA with technologies like artificial intelligence (AI) presents both challenges and opportunities [14]. Its real-time monitoring and safety protocols remain pivotal in contemporary industrial operations, shaping the future of automation [15].

PLC. A PLC is an industrial digital computer tailored for the control of manufacturing processes. They serve as the core control unit within the ICSs, executing control logic and coordinating the operation of different components. PLCs receive input from sensors, process it using pre-defined algorithms, and send commands to the actuators to control the physical processes. They use various communication protocols, such as Modbus, Profibus, or Distributed Network Protocol Version 3 (DNP3), to interact with sensors, actuators, and other components within the ICSs [9]. PLCs also interpret input signals from devices (like sensors) and produce output signals to control actuators, motors, valves, and other end devices. They form the core of many control process loops, ensuring the desired outcomes are achieved based on the input data and the logic programmed into them.

HMI. HMI serves as the interface between operators and the ICSs, providing a graphical representation of industrial processes. Operators can interact with the system, monitor alarms, and control process parameters through HMIs, gaining a comprehensive view of the operations and facilitating informed decision making. They communicate with PLCs using protocols such as Open Platform Communication (OPC), Modbus Transmission Control Protocol (TCP), or Ethernet/IP [16].

MODBUS. Modbus is one of the most widely used communication protocols in ICSs [6]. It is a serial communication protocol that enables communication between PLCs, RTUs, and other devices in the ICSs. Modbus supports both master–slave and client–server communication models. It is simple, lightweight, and easy to implement, making it a popular choice for industrial applications.

2.2. Related Work

The reviewed literature presents a multifaceted view of ICS security, encompassing various aspects such as protocol vulnerabilities, attack vectors, and defensive strategies, all of which are crucial for understanding the challenges identified in Section 1 for validation. For example, ref. [6]'s exploration of Modbus TCP vulnerabilities aligns with our research motivation to address the heightened risks in interconnected ICS environments. However, the study's focus on simulated scenarios raises questions about real-world applicability, echoing our research problem of bridging theoretical knowledge and practical reality.

The works of [9,17] contribute significantly to understanding DoS attacks and ICS testbed enhancements, respectively. While these studies align with the scope of our research in simulating attack scenarios, they reveal limitations in addressing the comprehensive nature of ICS threats and the unique configurations of these systems to validate cyber incidents and provide proper countermeasures.

The authors of [18] conducted a Man-in-the-Attack (MITM) attack to validate the designed testbed. In this attack scenario, the malicious actor assumes the role of an intermediary between the master and the control IED, impersonating either the data source or the data target. This deceptive tactic leads both the master and the control IED to believe they are engaged in direct communication. Once the connections are intercepted, the attacker gains complete control over the transmitted messages, allowing for manipulation or delay of the payload. This process exploits vulnerabilities such as CWE-287 (Improper Authentication), CWE-319 (Cleartext Transmission of Sensitive Information), CWE-306 (Missing Authentication for Critical Function), CWE-294 (Authentication Bypass by Capture-replay), and CWE-269 (Improper Privilege Management).

Article [19] introduces a real-time cyber–physical testbed and proceeds to conduct MITM and DoS attacks to validate the effectiveness of the testbed. The authors execute ARP cache poisoning to position themselves effectively between the network transmission path,

allowing them to intercept ongoing traffic between the master and the slave. Subsequently, the attacker impersonates the Modbus master to issue a Modbus command, triggering the slave IED. This attack exploits vulnerabilities such as CWE-319, CWE-287, and CWE-306. Additionally, a TCP SYN flood DoS attack is launched to overwhelm the device's resources, leveraging the inherent vulnerabilities of the Modbus protocol, including CWE-122 (Heap-based Buffer Overflow) and CWE-120 (Classic Buffer Overflow).

Further, the behavioural analysis approach in [20] resonates with our aim to delve into ICS-specific IOCs, though it primarily focuses on network anomalies and overlooks broader threat vectors. In [21]'s investigation of Modbus Flooding Attacks, we see an alignment with the challenges identified in Section 1, specifically the dynamic nature of ICS threats. However, their limited scope in addressing the wide spectrum of ICS vulnerabilities points to the research gap our study aims to fill.

The authors of [22] explored the enhancement of an established testbed for security validation in ICS. They developed a tool to enrich security datasets, enabling the testing, validation, and user training of integrated monitoring systems. The testbed can gather control system security data using elastic, analysing tactics and techniques from the MITRE ATT&CK framework. Lastly, ref. [23]'s study on mitigation strategies against attacks on the Modbus protocol offers insights into defensive mechanisms but also underscores the need for comprehensive strategies that are yet to be fully explored. Table 1 shows a review of the related literature.

Table 1. Literature review and related work.

Author(s)	Focus Area	Key Contribution	Limitations	Alignment with Research Scope
[6]	Modbus TCP vulnerabilities	Highlighted susceptibility to cyberattacks	Focus on simulated environments; questions real-world applicability	Addresses risks in interconnected ICSs
[13]	Enhancement of ICSs testbeds	Development of ICS datasets for security analysis	Limited by existing frameworks of testbeds and datasets	Supports the need for empirical research
[17]	Testbed validation with MITM attack	Demonstrated effectiveness of testbed through MITM attack	Not provided	Validates testbed under real-world attack scenarios
[18]	Real-time cyber-physical testbed	Conducted MITM and DoS attacks to validate testbed effectiveness	Not provided	Validates testbed in real-world scenarios
[19]	DoS attacks on PLCs	Analysed impact of DoS attacks; importance of availability in ICSs	Limited in addressing comprehensive vulnerabilities	Relates to simulating attack scenarios
[20]	Threat behaviours in ICSs	Shift from anomaly detection to behaviour-based analysis	Focuses mainly on network anomalies	Aligns with exploring ICSs-specific IOCs
[21]	Modbus flooding attacks	Effectiveness of anomaly-based detection algorithms	Does not address the broader spectrum of ICS vulnerabilities	Echoes dynamic nature of ICS threats
[22]	Testbed for ICS security validation	Development of ICS dataset validation tool	Limited by existing frameworks of testbeds and datasets	Relates to the simulation of large-scale scenarios
[23]	Mitigation strategies (Modbus)	Integration of security measures across OSI model layers	Highlights lack of comprehensive strategies for Modbus security	Points to the need for holistic security approaches

2.3. Addressing the Research Gap

The existing literature, while extensive, exhibits a gap in creating a holistic view of ICS cybersecurity that integrates the understanding of unique IOCs, real-world attack simulations, and practical validation methods. This gap is particularly pronounced in the context of the increasing interconnectivity and specialised nature of ICSs, as outlined in Section 1. This paper, therefore, aims to bridge this gap by developing a practical and effective approach for ICS security validation. We leverage the MITRE ATT&CK Framework in conjunction with open-source security tools to model and validate two major attack scenarios (DoS attack and ARP poisoning attack) specifically for ICSs. This approach not only addresses the theoretical and practical dichotomy in current research but also aligns with the challenges and motivations identified in Section 1.2. It also helps safeguard ICS infrastructure against evolving cyber threats.

3. System Model, Attack Model, and Capabilities

3.1. System Model

The proposed system model integrates a PLC (OpenPLC), a traffic light system as a slave device, and a SCADA system, specifically ScadaBR, to establish a comprehensive and interactive control environment. This model is designed to simulate real-world ICS operations and interactions, with a focus on examining the efficacy of control logic under varying conditions. Each plays a distinct role in the operational simulation, as follows:

OpenPLC_v3. Serving as the core of the system, OpenPLC represents the programmable logic controller. It is built on an open-source platform, offering significant advantages in terms of customisability, community-driven updates, and cost-effectiveness. OpenPLC is responsible for executing predefined logic control, thereby dictating the behaviour of the traffic light system [10].

Arduino traffic light system (Slave Device). The traffic light system, functioning as a slave device, provides observable outputs based on the control logic executed by the OpenPLC. Its behaviour offers insights into the accuracy and reliability of the control commands issued by the OpenPLC.

ScadaBR system. The ScadaBR system acts as the HMI, enabling real-time monitoring and, if necessary, manual intervention by operators. It provides visual feedback on the traffic light system's state, allows for data logging and analysis, and facilitates operator control when manual overrides are required [6].

3.1.1. Communication and Interaction

Communication between these components is orchestrated using the Modbus protocol. The OpenPLC sends and receives data to and from the traffic light system over Modbus. The ScadaBR system receives the control commands from the OpenPLC and allows the operator to issue these commands, which are then transmitted back to the OpenPLC for execution.

3.1.2. Traffic Light Control Logic

The traffic light system's response to the control logic provides a practical framework for evaluating the efficacy of the PLC's command execution and the overall system's reliability and safety.

The operation of the traffic light control system is articulated through a series of timers, triggers, and logic operations, ensuring safe and sequential transitions between light states, as shown in Table 2. Key elements include:

- Timers of flight (TOF0, TOF1, TOF2): these provide delays for transitions between light states.
- Falling edge triggers: these detect transitions from ON to OFF in light-emitting diodes (LEDs), initiating subsequent light changes.
- LEDs control logic: incorporates red, orange, and green LEDs, each activated under specific conditions to ensure a smooth transition between light states.

- Safety and diagnostic features: includes a feature to detect if no lamps are active for over a second, indicating potential malfunctions.
- Startup and initialisation: a timer at startup (TON0) stabilises the system before LED operations commence.
- Execution cycle configuration: the PLC's execution cycle is set to a 20-millisecond interval, ensuring timely responses and continuous condition evaluation.

Table 2. Arduino traffic light system diagram settings.

Point Name	Type	Location	Status	Value
RED_Light	Boolean	%Q×100.0	ON	True
YELLOW_Light	Boolean	%Q×100.1	ON	True
GREEN_Light	Boolean	%Q×100.2	OFF	False

3.2. Attacker Model and Capabilities

This paper's examination of attacks is based on specific assumptions that define the attacker model. In this attack model, we assume that the adversary has successfully infiltrated the target environment by employing social engineering techniques. The attacker has acquired unauthorised remote access privileges within the OT network. However, we assume that the attacker's knowledge of the system model gives them the capacity to perform the following attacks:

- DoS on Modbus Protocol.
- ARP poisoning attack on SCADA and PLC communication.

These skills will be further detailed and categorised by applying the MITRE ATT&CK framework.

The tools used in these attacks are:

1. Hping3. Hping3 is a versatile network tool used for network scanning, packet crafting, and firewall testing. It enables the customisation and transmission of Internet Control Message Protocol (ICMP), User Datagram Protocol (UDP), and TCP packets; hence, it is valuable for network diagnostics and security assessments [11].
2. Kali Linux. The Kali operating system (OS) is a renowned penetration testing and ethical hacking distribution equipped with a comprehensive toolkit for security assessments, vulnerability testing, and ethical hacking. It is widely utilised by cybersecurity professionals and researchers [10].
3. Ettercap. Ettercap is a comprehensive suite for MITM attacks on a Local Area Network (LAN). It features sniffing live connections, content filtering on the fly, and many other interesting tricks. It supports an active and passive dissection of many protocols and includes various features for network and host analysis. Typically used for ARP poisoning attacks, Ettercap can intercept data on a network and potentially modify traffic, making it a powerful tool for network security assessments and for understanding network vulnerabilities [24].
4. Wireshark. Wireshark is a prominent network protocol analyser. It can be used to monitor, capture, and analyse Modbus protocol communication, assisting in identifying network anomalies, troubleshooting issues, and enhancing the security and reliability of ICS networks. This makes Wireshark an invaluable asset for maintaining efficient and secure industrial operations [7].

3.3. Description of Attacks

This section describes the two forms of attacks, as shown in Figure 2. The attack scenarios presuppose that the intruder has obtained access to the ICS network discussed in Section 3.1 through a social engineering technique; hence, the attacker has the knowledge and full capability to perform these two predominant attacks, the DoS attack and the MITM attack [25,26], as shown in Sections 3.3.1 and 3.3.2.

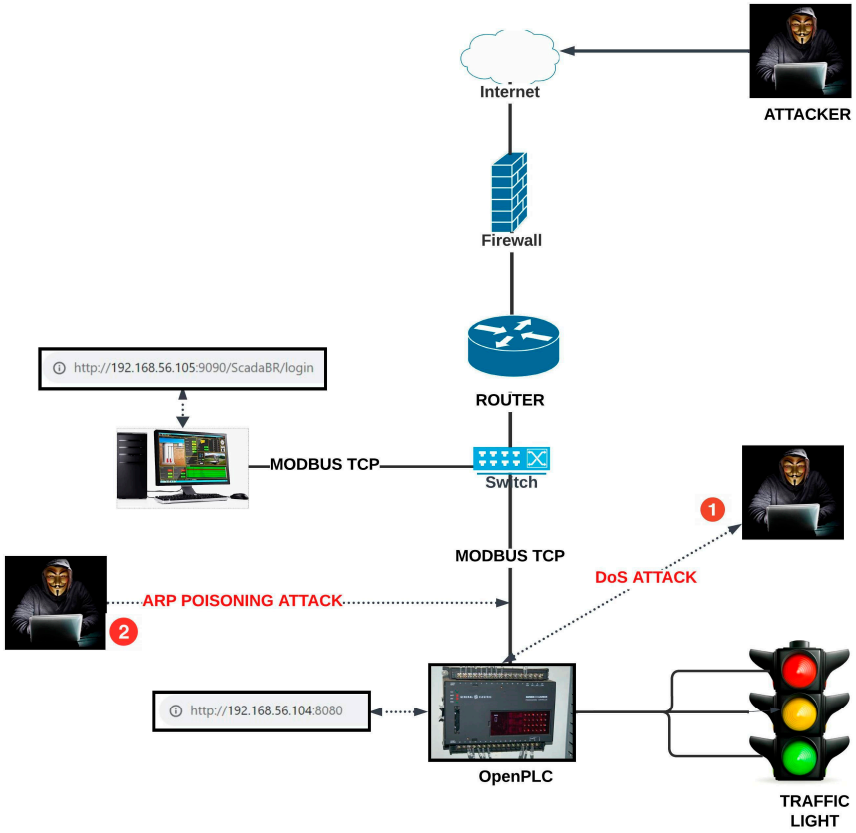


Figure 2. Attack model.

A DoS attack involves an attacker flooding a system or network with excessive traffic or malicious activity to disrupt its normal functioning, causing it to become inaccessible to legitimate users. In an MITM attack, an attacker inserts itself into communication between two devices with the intention of either listening in on the conversation or pretending to be one of the devices, creating the illusion of a legitimate exchange of information.

3.3.1. Attack Scenario I

In this attack scenario, the attacker aims to disrupt the normal operation of the PLC by flooding it with a high volume of spoofed packets. The objective is to overwhelm the PLC's processing capabilities, rendering it unable to handle legitimate requests, thereby causing a denial of service. The attack indicators and IOCs are shown in Tables 3 and 4.

Table 3. DoS attack indicators.

Indicator	MITRE ATT&CK Tactic	MITRE ATT&CK Technique	Description
High volume of incoming network packets to the PLC on port 502	Impact	Network Denial of Service (T1498)	The attacker is sending an overwhelming number of packets to the PLC, aiming to exhaust its resources.
Continuous traffic from randomised source IP addresses	Defence Evasion	Masquerading (T1036)	The attacker employs IP address spoofing to obfuscate their true origin, resulting in traffic from seemingly random IP addresses.
Unusual or malformed Modbus packets	Initial Access	Exploit Public-Facing Application (T1190)	The attacker crafts and sends malicious packets designed to exploit vulnerabilities in the Modbus protocol.
Anomalies in PLC behaviour or control decisions	Impact	Impact on Business. Service stop (T1489)	The PLC might make unexpected control decisions or fail to process genuine commands due to the flood of packets.

Table 4. DoS attacks' IOCs.

IOCs	Recommended Measures	Security Components	Validation Tool(s)
High volume of incoming network packets to the PLC on port 502	Implement continuous network traffic monitoring and analysis to identify unusual patterns. Implement rate limiting for incoming Modbus requests.	Intrusion Detection Systems (IDS)	Suricata and Wireshark
Continuous traffic from randomised source IP addresses	Continuously monitor network traffic for patterns indicating IP address spoofing. Employ ingress and egress filtering to block traffic with spoofed or suspicious source IP addresses.	Network Monitoring Tools, Firewalls	Wazuh security information and event management (SIEM), Suricata
Deep packet inspection, signature-based detection	Deploy deep packet inspection to analyse Modbus traffic. Use signature-based detection to identify malicious Modbus packets.	IDS	Suricata, Wireshark
Anomalies in PLC behaviour or control decisions	Implement SIEM solution for centralised event management. Configure rules and alerts for security incidents. Correlate events from multiple systems for threat detection.	Anomaly Detection Systems	Wazuh SIEM, PLC Logs

To execute this attack, the attacker crafts a script command that exploits vulnerabilities in the Modbus protocol, which typically operates on TCP port 502. This script is designed to continuously send a barrage of TCP packets to the PLC's communication interface. Given that the PLC is expecting Modbus packets on port 502, it will try to process each incoming packet. However, since these packets are maliciously crafted and sent in rapid succession, the PLC becomes overwhelmed, exhausting its computational resources.

The packets sent by the attacker might have randomised source Internet Protocol (IP) addresses, making them challenging to trace back to the actual origin of the attack. This technique, known as IP address spoofing, further complicates the mitigation process. As the PLC struggles to process the flood of incoming packets, its responsiveness to genuine Modbus commands diminishes, leading to significant operational downtime and potentially jeopardising the safety and functionality of the system.

3.3.2. Attack Scenario II

In this attack scenario, an adversary seeks to exploit the inherent trust between the SCADA system and the PLC. The goal is to clandestinely hijack the communication channel, positioning themselves as an invisible conduit in the data exchange, unbeknownst to either entity. Such a stealthy manoeuvre allows the attacker to eavesdrop on the ongoing communications. Tables 5 and 6 show the attack indicators and the attack IOCs respectively.

The attack unfolds in stages. It begins with an ARP cache poisoning technique. Here, the attacker leverages tools like Ettercap to poison the ARP cache of both the SCADA system and the PLC. This deceit ensures that both entities mistakenly recognise the attacker's machine as the legitimate communication partner. Consequently, all Modbus/TCP packets intended for the SCADA system or the PLC are inadvertently channelled through the attacker's machine.

Table 5. ARP poisoning attack indicators.

Indicator	MITRE ATT&CK Tactic	MITRE ATT&CK Technique	Description
ARP Traffic Anomalies	Discovery	Network Service Scanning (T1046)	A surge in ARP traffic indicates attempts to identify devices on the network by sending ARP requests.
Media Access Control (MAC) Address Discrepancies in ARP tables	Defence Evasion	Spoofing (T1564)	Manipulated ARP tables show the attacker's MAC address associated with a legitimate IP.
Modbus/TCP Traffic Irregularities	Collection	Network Traffic Capture (T1042)	Intercepted or altered Modbus/TCP packets lead to unexpected network traffic patterns.
Presence of ARP Spoofing Tools	Discovery	System Network Configuration Discovery (T1016)	Tools like Ettercap on the network signify ARP cache poisoning attempts.

Table 6. ARP poisoning attacks' IOCs.

IOCs	Recommended Measures	Security Components	Validation Tool(s)
ARP Traffic Anomalies	Implement continuous network traffic monitoring and analysis.	IDS	Snort/Wireshark
Deep Packet Inspection, Signature-Based Detection	Deploy deep packet inspection to analyse packet payloads. Use signature-based detection for known attack patterns.	IDS	Snort
MAC Address Discrepancies in ARP Tables	Continuously monitor ARP tables for discrepancies. Validate MAC address associations with IP addresses.	Network Monitoring Tools	Arptables
Modbus/TCP Traffic Irregularities	Implement deep packet inspection for Modbus traffic. Analyse Modbus packet anomalies.	IDS	tcpdump, Wireshark, Snort
ARP Spoofing Detection	Deploy intrusion detection systems that can detect ARP spoofing attempts. Regularly scan for the presence of ARP spoofing tools.	IDS	Snort

Having discreetly established this MITM position, the attacker gains an unobstructed view of the Modbus/TCP packets shuttling between the SCADA system and the PLC. As the packets transit through the attacker's machine, every piece of information, command, and response becomes visible to the attacker, offering an unparalleled insight into the operations and communications between the SCADA and the PLC.

4. Our Approach

Security validation in ICSs is instrumental for comprehending the behaviour of assets and systems under diverse forms of attack as well as empowering proactive responses. Our proposed approach is designed to assist industrial users, including system operators and cybersecurity engineers, in comprehending, identifying, and proactively responding to DoS attacks and ARP poisoning attacks on ICSs within their operational environment, thereby preventing potential damage.

In this work, we utilised open-source tools to showcase the practical application of our approach. The obtained results will equip operators with valuable insights to make informed and proactive decisions, such as to swiftly identify and address potential threats, minimising the risk of damage. Beyond its immediate practical implications, our work will also contribute to the broader academic research landscape in ICS security validation

by showcasing the effective utilisation of open-source tools in validating IOCs for specific attack scenarios.

4.1. Experimental Design

The experimental environment is a controlled virtual environment that includes the following components: a Windows 10 (64-bit) machine, an Ubuntu 22.04 server, OpenPLC_v3, ScadaBR, a Debian server, and a Kali Linux virtual machine. The design is tailored to meet the research objective; therefore, all the components in the design communicate effectively with each other. Table 7 shows the experimental configuration of all this equipment.

Table 7. Experimental equipment configuration.

Equipment/Component	System/Software/Version	IP Address
Windows 10vm	Windows 10 (64-bit), Base Memory: 2048 MB.	192.168.56.105
ICS Project	Ubuntu (64-bit), Base Memory: 2096 MB, Processors: 2	192.168.56.107
Kali-linux-2022.4-virtualbox-amd64	Kali-Linux-2022.4-virtualbox-amd64 Memory: 2048 Processors: 2	192.168.56.102
Wazuh v4.5.1 Ova	Wazuh v4.5.1 Ova, Linux 2.6 /3.x/4.x (64-bit), Base Memory; 2313 MB, Processors: 2	192.168.56.106
Suricata	7.1	192.168.56.107
Snort	Snort3	192.168.56.107
Wazuh	4.5.1	192.168.56.106

- Windows 10 (64-bit) Virtual Machine (IP: 192.168.56.105): the Windows server is a virtual box and hosts the ScadaBR application, which serves as the HMI for monitoring and controlling the traffic light system.
- Ubuntu 22.04 virtual server (IP: 192.168.56.107): The Ubuntu machine hosts the OpenPLC-v3, which is responsible for controlling the slave device. Additionally, the snort_v3 IDS and Suricata IDS are also installed on the Ubuntu server for monitoring network traffic for suspicious activities.
- Debian OS (IP 192.168.56.106): Debian OS is deployed within the network to support additional security and monitoring components. It is dedicated to hosting and configuring the Wazuh SIEM solution.
- Kali Linux Virtual Machine (IP 192.168.56.102): Since the attacker is assumed to already have access to the ICS environment and have full knowledge of the system model, it is represented as a Kali virtual and has been added to the network. It has been configured with all the necessary tools to aid in launching the attacks on IP 192.168.56.102.

ScadaBR. ScadaBR is an open-source SCADA system that provides an HMI for real-time monitoring and control of industrial processes. It supports remote monitoring, customisation, and real-time data visualisation. The host IP for the ScadaBR is 192.168.56.105 and it has been configured to communicate with the OpenPLC on Modbus communication protocol port 502 as well as provide a watch view of the traffic light system. As part of the configuration, we accessed the ScadaBR from the web browser on the Windows 10 VM on port 9090 and carried out the following configurations, as shown in Figure 3.

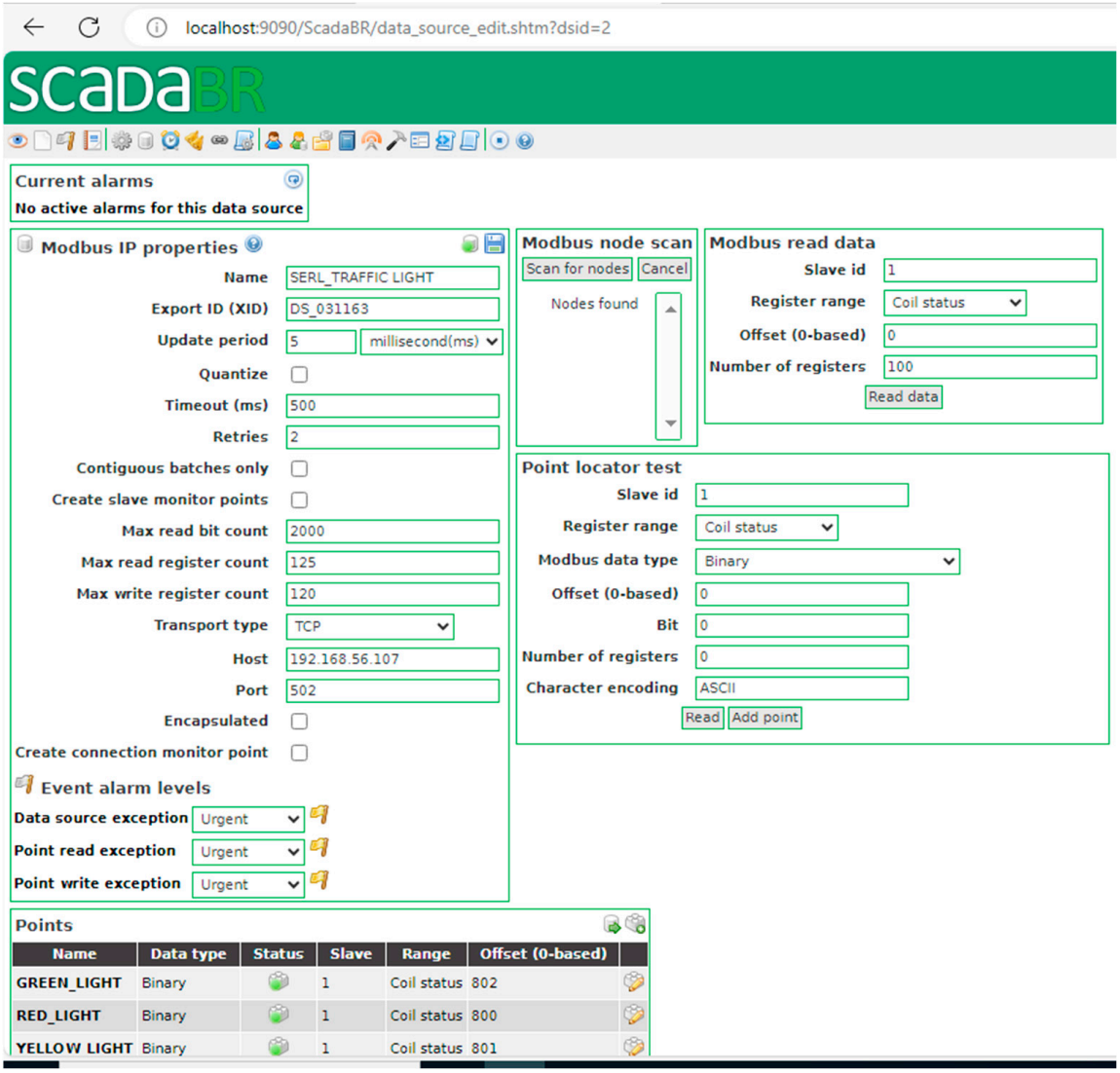


Figure 3. Experimental ScadaBR configuration.

- We assigned a name to our data source connection as “serl traffic light” and the connection type as “Modbus IP”.
- Set the update period to “500 ms” for real-time updates.
- Set the timeout to “500 ms”.
- Transport type to “TCP with keep-alive”.
- Host IP address to the OpenPLC runtime IP: “192.168.56.107”.

OpenPLC: OpenPLC is an open-source initiative that provides software and hardware blueprints for implementing PLCs [10]. Just like traditional PLCs, OpenPLC can be used in control loops to monitor and control processes. Its open-source nature allows users to customise, modify, and extend its functionalities as per their specific needs. In this paper, OpenPLC is responsible for controlling the slave device (traffic light), executing ladder logic (explained in Table 1), and communicating with ScadaBR on the Modbus protocol. The host IP of OpenPLC is 192.268.56.107. On the host machine, the OpenPLC runtime can be assessed on port 8080 from a web browser on the host virtual machine after the runtime time has been started from the command line. The physical connection of these devices is shown in Figure 4.

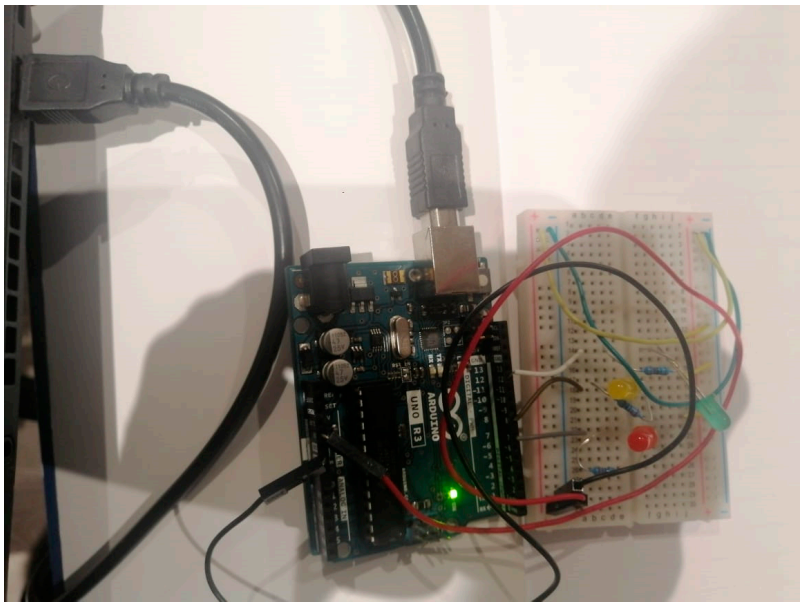


Figure 4. Experimental hardware equipment with connections.

4.2. Overall Execution

When our attack is simulated from the attacker's machine, the network IDS (NIDS)'s open-source tool equipped with predefined rules for detecting IOCs will promptly flag and log the identified attack. Subsequently, the logs will be transmitted to our SIEM solution, which, in turn, will process the data, trigger alerts, and visually present the information on a dashboard. This process enables a proactive response by the system operator, ensuring timely awareness and swift actions to mitigate potential threats.

5. Results

This section presents the open-source tools deployed for the security validation of the IOCs shown in Tables 3–6 of the two simulated attack scenarios, as well as the findings, results, and evaluation of the security validation in ICSs.

5.1. Open-Source Tools

5.1.1. Network Intrusion Detection System

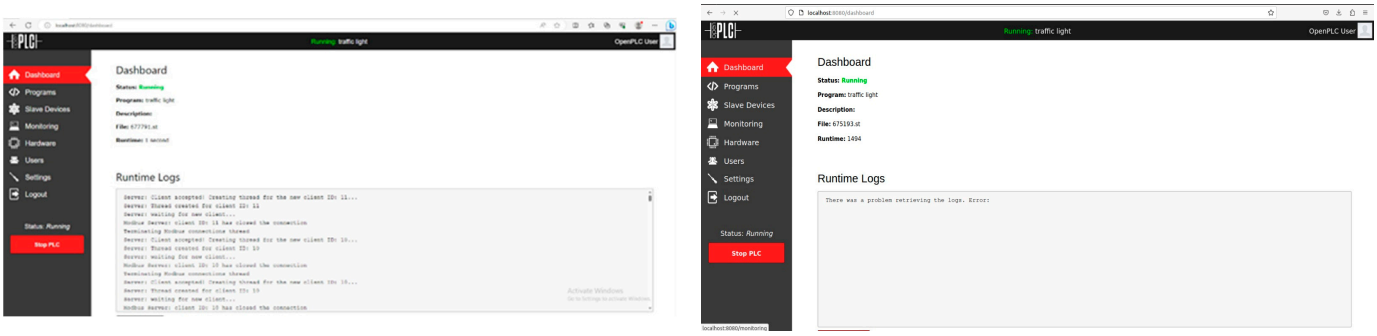
- Snort. Snort IDS is a widely used open-source IDS. It is designed to actively monitor network traffic for intrusion detection and the prevention of malicious network activities in real time [27]. Utilising its signature engine, continuous community backing, and ongoing development efforts simplifies the deployment of new detection rules.
- Suricata. Suricata is an open-source Network IDS, IPS, and Network Security Monitoring (NSM) engine. It is known for its capability to provide real-time intrusion detection, which is essential for monitoring and safeguarding network traffic against malicious activities.

5.1.2. SIEM, Dashboard, and Visualisation Utilisation

Wazuh is equipped with pre-configured settings and data parsers for various security tools, including our IDS tools, Snort, and Suricata, to process alerts and logs efficiently. We configured and integrated Snort IDS and Suricata IDS. The alerts and logs from Suricata and Snort are directed to the Wazuh agent and manager for further handling. The Wazuh manager correlates these alerts and system logs with other security events, enforces additional security rules, and triggers custom responses based on the received data. Wazuh plays a crucial role in providing a comprehensive view of the security status of the ICS environment. It also has a dashboard capability, which we leveraged to visually represent security events and alert trends, aiding in quick identification and validation.

5.2. DoS Attack Validation (Attack Scenario I)

The DoS attack destination IP and port address are 192.168.56.107:502. First, using hping3 to launch our attack, a marked increase in network traffic directed towards the PLC's Modbus port was observed, signifying the success of the scripted DoS attack. This surge in traffic disrupted the performance of the OpenPLC and monitoring, as shown in Figures 5 and 6. Using Wireshark and tcpdump, the Modbus traffic was effectively captured and logged, as shown in Figure 7.



(**a**) (**b**)

Figure 5. State changes in the OpenPLC logs before and after the DoS attack. (**a**) Pre-DoS attack; (**b**) post-DoS attack.

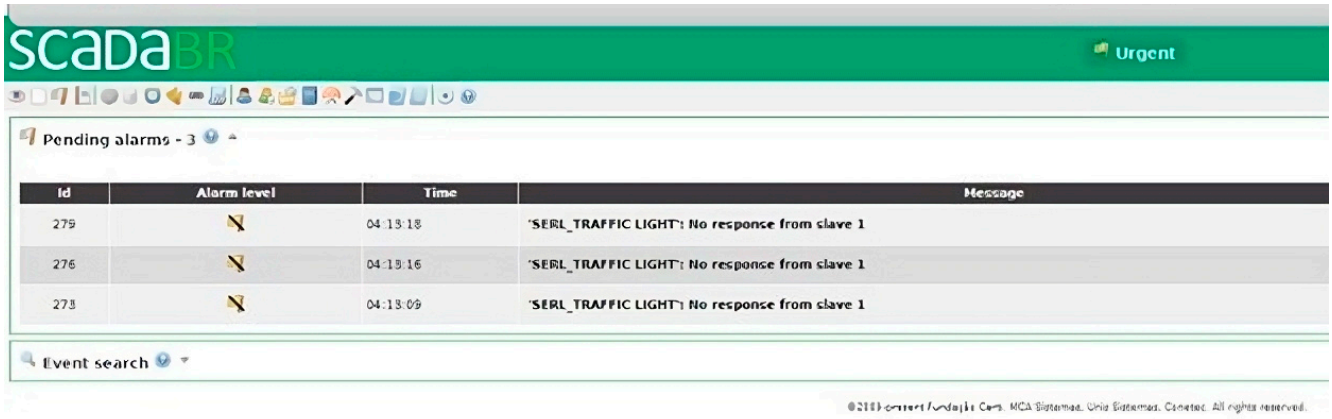


Figure 6. ScadaBR connection during a DoS attack.

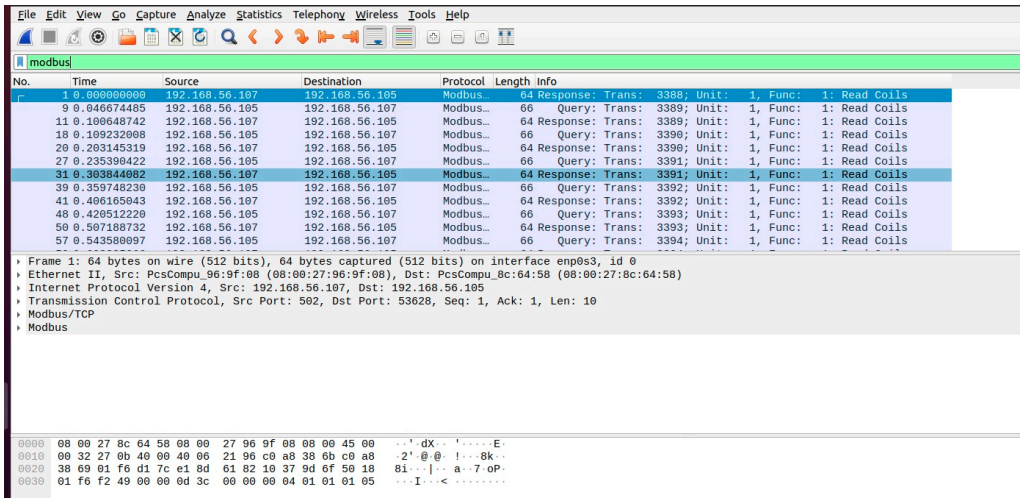


Figure 7. Wireshark captures traffic during a DoS attack.

5.2.1. Suricata Performance

Suricata played a pivotal role in detecting this abnormal spike in network traffic after the execution of the attack code "hping3 -d 120 -s -w 64 -p 502 –flood –rand-source –interval u1000 192.168.56.107" from our Kali machine. It effectively flagged these patterns as potential DoS attack activities, aligning with the pre-defined characteristics and rules. It also flagged the other intriguing facet of ever-shifting source IPs from the randomness and broad spectrum as indicative of IP spoofing, a tactic frequently adopted in advanced DoS onslaughts as logs, as shown in Figure 8.

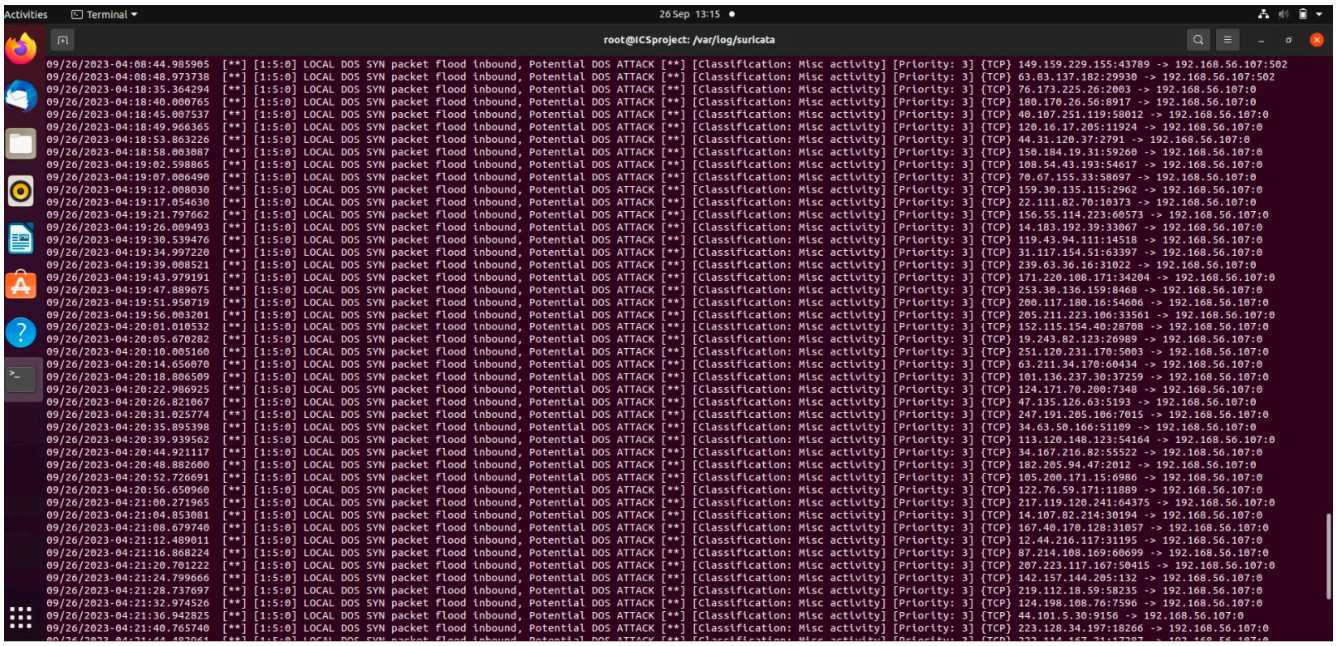


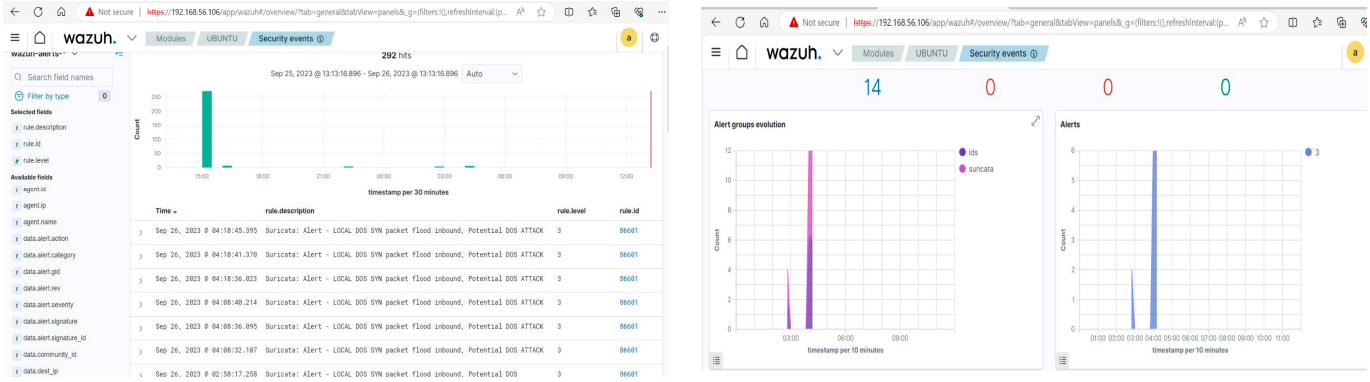
Figure 8. Suricata DoS alert logs.

The detailed explanation of the command "hping3 -d 120 -s -w 64 -p 502 –flood –rand-source –interval u1000 192.168.56.107" is as follows:

- -d 120: Set the byte size of the data section (payload) of the packet to 120 bytes.
- -s: Spoof the source address (make it look like the packets are coming from a different IP).
- -w 64: Set the TCP window size to 64.
- -p 502: Target port 502.
- --flood: Send packets as fast as possible without showing any output.
- --rand-source: Randomise the source IP address.
- --interval u1000: Send one packet every 1000 microseconds.
- 192.168.56.107: The target IP address.

5.2.2. Wazuh Alert Correlation

Integrating the alerts into the Wazuh SIEM platform provided a multi-dimensional view of the attack. The platform aggregated and correlated the alerts, painting a comprehensive picture of the attack timeline and its attributes. Notably, the attack signatures recognised by the Wazuh were consistent with known DoS attack profiles within the MITRE ATT&CK framework, as shown in Figure 9.



(**a**) (**b**)

Figure 9. DoS attack alerts on the Wazuh dashboard. (**a**) DoS attack alert; (**b**) DoS attack alert group.

5.3. ARP Poisoning Attack Validation (Attack Scenario II)

The ARP poisoning attack discussed in Section 3 aimed at compromising the communication channel between the SCADA system and the PLC. The results are as follows:

A rerouting of Modbus/TCP packets through the attacker's machine was successfully detected, indicating the successful execution of the ARP poisoning attack code "sudo ettercap -T -i eth0 -M arp: remote /192.168.56.107// /192.168.56.105//" from the attacker's machine. The use of ettercap for this purpose proved to be highly effective. We extracted the ARP tables of the ScadaBR system and OpenPLC, as shown in Figure 10. The analysis of these tables, especially the MAC addresses, showcased noticeable inconsistencies. The MAC addresses, which typically align with recognised devices, had now been altered to point towards the attacker's machine.

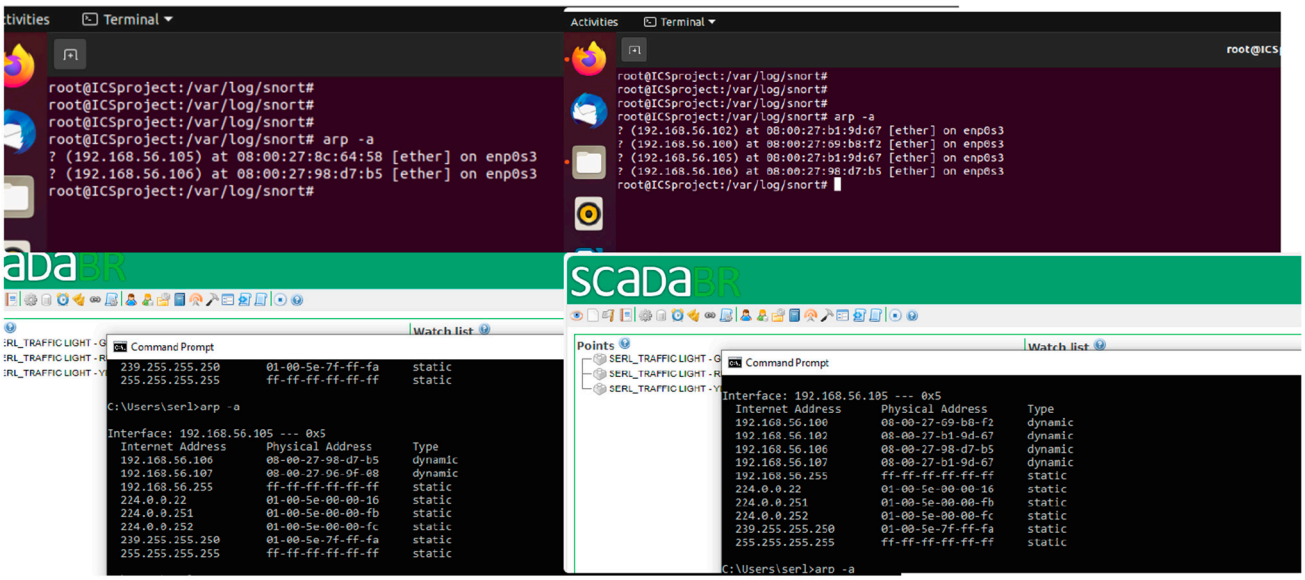


Figure 10. ARP tables.

Network Behaviour Analysis

Substantial anomalies in the network behaviour were identified and logged by our Snort IDS tools and the Wazuh SIEM solution. These indicators of the various stages of the ARP poisoning attack validate the effectiveness of the techniques and tools employed in identifying the security breach.

Snort monitored and logged the irregular and unexpected ARP broadcast, its frequency, and the uncharacteristic MAC address changes. The ability to discern these subtle yet

critical variations in network traffic provided early warning signs for the ARP, as shown in Figure 11.

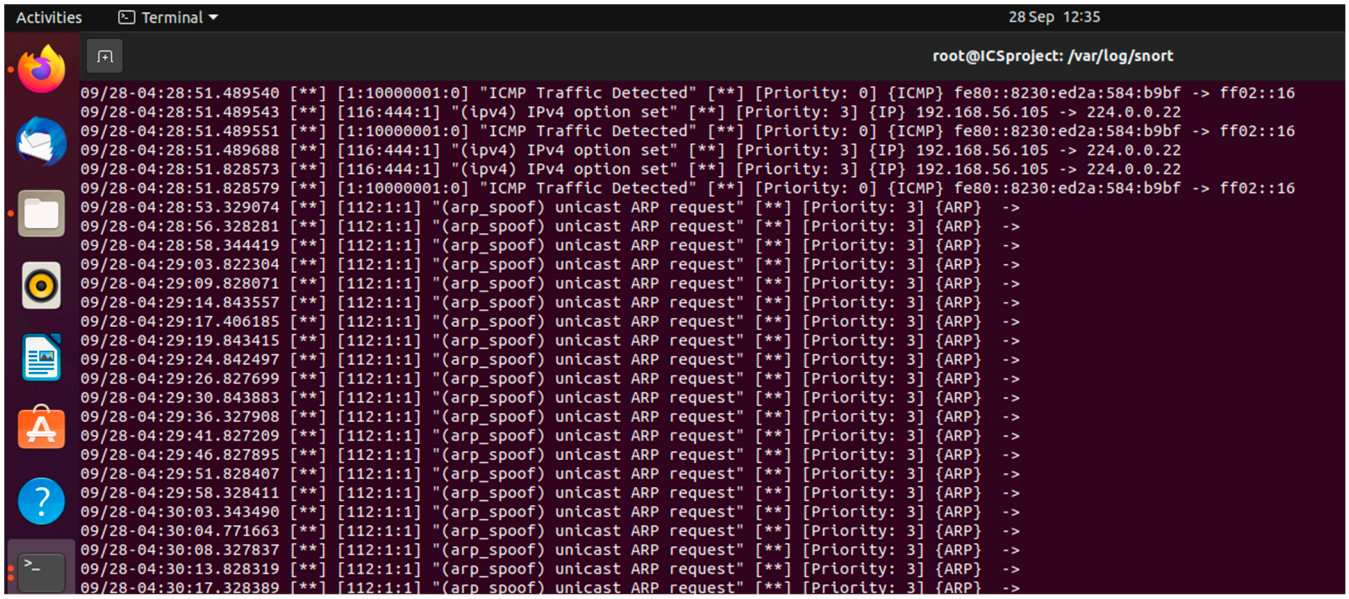


Figure 11. Snort ARP alert logs.

6. Conclusions

The work described in this paper has explored the area of cyber-attack validation in ICSs. We have validated two common attacks, DoS and ARP poisoning attacks, in ICSs.

The techniques deployed successfully addressed the attacks mentioned in the threat model. The accurate detection and timely validation of the DoS and ARP poisoning attacks indicate a high level of security efficacy. In terms of performance, the implementation of security measures showed a high degree of reliability. The tools used were able to validate the mapped IOCs of the ICS environment under test. The alignment with the MITRE ATT&CK framework further validates the security measures, confirming that the techniques are not only effective in a simulated environment but also applicable in real-world ICS scenarios. Key findings from the simulated attack scenarios demonstrate the effectiveness of the proposed approach.

The DoS attack validation revealed significant vulnerabilities within the PLC's Modbus protocol communication, a finding that underscores the importance of robust network traffic monitoring and intrusion detection systems. Similarly, the ARP poisoning attack validation highlighted the critical need for vigilant network behaviour analysis and the importance of regular checks for discrepancies in ARP tables. Therefore, the attack validation scheme proposed in this paper is useful for the future development and security of ICSs. The results not only validate the proposed methods but also provide actionable insights for ICS security enhancements.

However, the constantly evolving landscape of cyber threats, especially with the advent and incorporation of IoT devices, edge computing, and cloud services in ICSs, necessitates a broader exploration of potential threat avenues. Future work includes simulating and validating more attacks, such as malware attacks and supply chain attacks.

Author Contributions: Conceptualisation, D.S.A. and M.A.; methodology, D.S.A., M.A. and N.S.; writing—original draft preparation, D.S.A.; writing—review and editing N.S.; supervision, N.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data is contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Asiri, M.; Saxena, N.; Gjomemo, R.; Burnap, P. Understanding Indicators of Compromise against Cyber-attacks in Industrial Control Systems: A Security Perspective. *ACM Trans. Cyber-Phys. Syst.* **2023**, *7*, 1–33. [CrossRef]
2. Security Affairs. (14 March 2022). Anonymous Hacked German Subsidiary Rosneft. Security Affairs. Available online: https://securityaffairs.co/129052/hacktivism/anonymous-hacked-german-subsidiary-rosneft.html (accessed on 13 February 2024).
3. Railway Technology. Belarus: Hackers Attack Train Systems. 29 December 2023. Available online: https://www.railway-technology.com/news/belarus-hackers-attack-train-systems/ (accessed on 13 February 2024).
4. Huang, H.; Zhang, N.; Luo, X.; Xu, Y.; Xu, Z. A Survey on Threat Intelligence-driven Industrial Control System Security. *IEEE Trans. Ind. Inform.* **2021**, 1–20.
5. Ahmadi-Assalemi, G.; Al-Khateeb, H.M.; Epiphaniou, G.; Aggoun, A. Super Learner Ensemble for Anomaly Detection and Cyber-Risk Quantification in Industrial Control Systems. *IEEE Internet Things J.* **2022**, *9*, 13279–13297. [CrossRef]
6. Mohammed, A.S.; Anthi, E.; Rana, O.; Saxena, N.; Burnap, P. Detection and mitigation of field flooding attacks on oil and gas critical infrastructure communication. *Comput. Secur.* **2023**, *124*, 103007. [CrossRef]
7. Melamed, R.; Shabtai, A.; Elovici, Y. Adversarial Attacks on Industrial Control Systems: A Survey on Techniques and Mitigation Approaches. *IEEE Trans. Ind. Inform.* **2021**, *17*, 5719–5736.
8. Clarke, S. *SCADA: Supervisory Control and Data Acquisition*; ISA: Tokyo, Japan, 2013.
9. Stouffer, K. Cybersecurity in SCADA. *Cybersecur. Ind. Syst.* **2018**, *6*, 22–29.
10. Thompson, R. Centralized monitoring in SCADA systems. *IEEE Ind. Electron. Mag.* **2016**, *10*, 28–37.
11. Hughes, T. Data acquisition in SCADA. *Autom. Constr.* **2018**, *22*, 405–412.
12. Kumar, A. Alarm management in SCADA. *Process Control J.* **2019**, *12*, 44–50.
13. Choi, S.; Choi, J.; Yun, J.-H.; Min, B.-G.; Kim, H. Expansion of ICS Testbed for Security Validation based on MITRE ATT&CK Techniques. In Proceedings of the 13th USENIX Workshop on Cyber Security Experimentation and Test, Online, 10 August 2020.
14. Fernandez, G. SCADA and modern tech. *Ind. Inform. J.* **2021**, *18*, 39–46.
15. Gomez, J. SCADA's efficiency role. *Ind. Inform. Rev.* **2020**, *16*, 58–65.
16. Gharibi, R.H.; Zarei, A.M.; Khayyambashi, M.R. A Review on Security Validation Techniques for Industrial Control Systems. In Proceedings of the 2019 IEEE International Conference on Industrial Cyber-Physical Systems (ICPS), Taipei, Taiwan, 6–9 May 2019.
17. Banik, S.; Banik, T.; Hossain SM, M.; Saha, S.K. Implementing Man-in-the-Middle Attack to Investigate Network Vulnerabilities in Smart Grid Test-bed. In Proceedings of the 2023 IEEE World AI IoT Congress (AIIoT), Seattle, WA, USA, 7–10 June 2023.
18. Chen, B.; Butler-Purry, K.L.; Goulart, A.; Kundur, D. Implementing a Real Cyber-Physical System Test Bed in RTDS and OPNET. In Proceedings of the 2014 North American Power Symposium (NAPS), Pullman, WA, USA, 7–9 September 2014; pp. 1–6.
19. Yılmaz, E.N.; Ciylan, B.; Gönen, S.; Sindiren, E.; Karacayılmaz, G. Cyber Security in Industrial Control Systems: Analysis of DoS Attacks against PLCs and the Insider Effect. In Proceedings of the 2018 6th International Istanbul Smart Grids and Cities Congress and Fair (ICSG), Istanbul, Turkey, 25–26 April 2018.
20. Dragon, Inc. Mapping-Industrial-Cybersecurity-Threats-to-MITRE-ATTACK-for-ICS. Scribd. April 2020. Available online: https://www.scribd.com/document/505064681/Mapping-Industrial-Cybersecurity-Threats-to-MITRE-ATTACK-for-ICS (accessed on 25 February 2024).
21. Bhatia, S.; Kush, N.; Djamaludin, C.; Akande, J.; Foo, E. *Practical Modbus Flooding Attack and Detection*; Information Security Discipline, Queensland University of Technology: Queensland, Australia, 2019.
22. Dehlaghi-Ghadim, A.; Balador, A.; Moghadam, M.H.; Hansson, H.; Conti, M. ICSSIM—A framework for building industrial control systems security testbeds. *Comput. Ind.* **2023**, *148*, 103906. [CrossRef]
23. Rahman, A.; Mustafa, G.; Khan, A.Q.; Abid, M.; Durad, M. Comprehensive Analysis of Vulnerabilities in the Modbus Protocol and Exploitation through Denial of Service Attacks. *Int. J. Crit. Infrastruct. Prot.* **2022**.
24. Haitao, X.; Chen, Z.; Song, L.; Bo-Sian, J.; Zhigang, L.; Fei, W.; Yuling, L. CapsITD: Malicious Insider Threat Detection Based on Capsule Neural Network. In Proceedings of the International Conference on Security and Privacy in Communication Systems, Hongkong, China, 19–21 October 2023. [CrossRef]
25. Rajesh, L.; Satyanarayana, P. Detecting Flooding Attacks in Communication Protocol of Industrial Control Systems. *Int. J. Adv. Comput. Sci. Appl.* **2020**, *11*, 396–400.
26. Sahu, A.; Mao, Z.; Wlazlo, P.; Huang, H. Multi-Source Multi-Domain Data Fusion for Cyberattack Detection in Power Systems. *IEEE Access* **2021**, *9*, 2–20. [CrossRef]
27. Roesch, M. Snort: Lightweight Intrusion Detection for Networks. In Proceedings of the Large Installation System Administration Conference (LISA), Seattle, WA, USA, 7–12 November 1999; Volume 99, pp. 229–238.