encyclopedia

MDPI

Review

An Overview of the Security of Programmable Logic Controllers in Industrial Control Systems

Hui Cui [1,*], Jin Hong [2] and Rodney Louden [3]

[1] Faculty of IT, Monash University, Melbourne, VIC 3800, Australia
[2] Department of Computer Science and Software Engineering, University of Western Australia, Perth, WA 6009, Australia; jin.hong@uwa.edu.au
[3] Critical Infrastructure Technologies, South Fremantle, WA 6162, Australia; rodney.l@citech.com.au
* Correspondence: hui.cui@monash.edu

Abstract: One key role in industrial control systems (ICSs) is known as Programmable Logic Controller (PLC). However, with the development of the Internet of Things (IoT), PLCs have become exposed to an increasing number of attacks, which may cause malfunctions of the whole ICS. Thus, it is necessary to identify potential attacks on PLCs and propose effective solutions to mitigate them. Unfortunately, to date, there have not been significant efforts made to provide a detailed overview of existing works on PLC security. With such a concern in mind, in this paper, we focus on summarising PLC security from different components running at different layers of a PLC architecture. We first review the framework of PLCs; then, we discuss several models when considering PLC security. After that, we provide an overview of existing attacks on PLCs and general solutions to those issues from different perspectives. Lastly, we conclude this paper with an overview of future research areas in PLC security.

Keywords: automation; information control systems; programmable logic controllers; security; critical infrastructure

Citation: Cui, H.; Hong, J.; Louden, R. An Overview of the Security of Programmable Logic Controllers in Industrial Control Systems. *Encyclopedia* **2024**, *4*, 874–887. https://doi.org/10.3390/encyclopedia4020056

Academic Editor: Raffaele Barretta

Received: 2 April 2024
Revised: 13 May 2024
Accepted: 17 May 2024
Published: 22 May 2024

1. Introduction

In recent decades, Industrial Control Systems (ICSs) have been widely deployed to control and monitor operations of critical infrastructures, including transportation, power grids, and water treatment units [1,2]. In recent decades, due to the trend of connecting ICSs to the Internet, the security of ICSs has received significant attention. It has been estimated that the global ICS market will grow to $23.5 billion by 2026 [3]. While ICSs are transformed by smart Internet of Things (IoT) devices with increasing usability, efficiency, and productivity, Internet of Things (IoT) devices also significantly impact ICS security [4,5].

Programmable Logic Controllers (PLCs), along with sensors and actuators, are key components of ICSs, as ICSs are monitored and operated via PLCs. Traditionally, it is believed that PLCs are isolated from outside network connections, and thus, PLCs should not be infected by computer viruses. However, several incidents indicate that PLCs are at a significant risk despite them being separated from the core network. For example, a former employee hacked the Queensland computerized waste management system in 2000, which caused a large amount of sewage to be dumped into different areas of the city [6]. A malfunction caused by worms inside computers was detected in monitoring systems in the Ohio Davis-Besse nuclear plant in 2003 [7]. Nevertheless, these earlier incidents did not raise the scientific community's interest. It was not until recent years that security concerns in PLC-based automated systems started to attract public attention. In 2010, the Stuxnet virus was discovered in Iran's nuclear facilities [8]. After that, PLC producers and users began to identify vulnerabilities and explore countermeasures to these threats. In the past decade, there has been a large number of papers either focusing on potential attacks that can be launched against PLC-related systems or different prevention mechanisms

to mitigate various security issues in PLC-based systems. However, there has been little effort devoted to providing a complete overview covering all aspects of PLC security. In this paper, our focus is on providing an overview of existing security issues in PLCs and relevant techniques that can be applied to mitigate or prevent those potential attacks.

1.1. Related Works

There have been several papers focused on presenting a summary of PLC vulnerabilities and countermeasures. Basnight et al. [9] discussed the vulnerability of PLCs in terms of intentional firmware modifications to understand the feasibility of firmware modification attacks caused by threats in PLC firmware. Sandaruwan, Ranaweera, and Oleshchuk [10] presented several PLC vulnerabilities via various types of attack vectors affecting the critical infrastructure. Wardak, Zhioua, and Almulhem [11] conducted an investigation into PLC access control problems, especially with regard to the password-based access control. Ghaleb, Zhioua, and Almulhem [12] provided a security analysis over network communications between stations responsible for setup and configuration and PLCs. Serhane et al. [13] provided suggestions on policies, recommendations, and countermeasures to secure PLC-based systems. Wu et al. [14] summarized PLC security from perspectives including firmware security, operation security, and program security. Pan, Wang, and Sun [15] reviewed PLC security in terms of code security, firmware security, network attack, and MODBUS protocol security, as well as certain protection mechanisms.

Contributions in this paper are different from other existing review papers about PLCs in several aspects. Firstly, the majority of previous survey papers focus only on one issue in PLC-based systems rather than providing a complete picture of all problem types. Secondly, some survey papers which provide an overview of PLC security from different aspects fail to cover relevant papers discussing those specific issues. Thirdly, existing survey papers do not include threat models of PLC security. Considering the incompleteness of existing overviews of PLC security, in this paper, our focus is on summarizing the security of PLCs from a wider perspective to cover all aspects related to PLCs.

1.2. Organization

This paper's remaining sections are organized as follows. In Section 2, we briefly describe an overview of PLC architecture. In Section 3, we discuss different threat models of PLC security. In Section 4, we summarize different types of attacks on PLCs. In Section 5, we present several techniques to mitigate PLC threats. In Section 6, we predict future research areas in PLC security. Lastly, this paper is concluded in Section 7.

2. Background

PLCs are extensively used as field devices in ICSs. A PLC provides a user-programmable interface between physical inputs and outputs to support customized control of ICS components. In this section, we present an overview of the components of PLCs.

2.1. PLC Architecture

An ICS is composed of a plant that describes the physical process under control, a group of sensors that read the plant's states (e.g., pressure, temperature), convert the states into electrical signals and deliver these signals to the controller; one or more PLCs monitoring and controlling the plant's states, which read sensor signals, execute control methodologies, and send actuators the corresponding signals; and a set of actuators receiving signals from PLCs and change the plant's states. ICSs are managed in a centralized control manner, and thus, PLCs are connected to a human–machine interface (HMI) which is a central control terminal operated by a human operator to manage the system (together with Supervisory Control and Data Acquisition (SCADA) [16]) as in Figure 1. The operator is able to remotely supervise the PLCs as well as applications running on PLCs. Every application's information can be loaded from a PLC, including the source code and the metadata information.

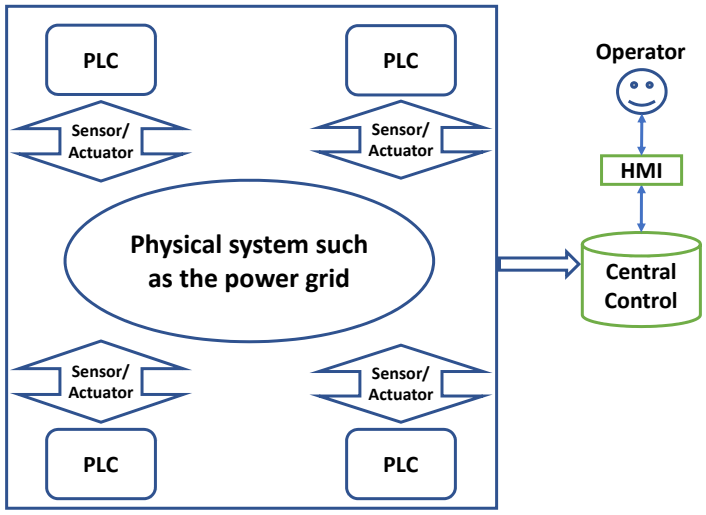

Figure 1. PLC architecture.

PLCs are increasingly connected to the Internet. As a result, PLCs are expected to support various network protocols. A variety of protocols are used by PLCs to communicate with other devices in ICSs, including Ethernet/IP [17], MODBUS [18], Distributed Network Protocol (DNP3) [19], Message Queuing Telemetry Transport (MQTT) [20], Open Platform Communications Unified Architecture (OPCUA) [21], and so on. Unfortunately, these protocols were not designed with security properties because security was not a concern when these protocols were first introduced to ICSs.

Following the work in [22], PLCs can be described as having three operational layers: the programming layer, the firmware layer, and the hardware layer.

2.2. Hardware Layer

The hardware layer in PLCs comprised electronic elements and microchips. Its key components include non-volatile storage, volatile memory, and microprocessors. The hardware layer is subject to attacks ranging from the physical control of the hardware, hardware design flaws bringing the software exploitation to the compromise of the supply chain. For the physical manipulation, it requires the attacker to have access to the device, and thus it is a least likely attack and probably involves a malicious insider. Software exploitation of hardware flaws relates to exploits to the layer of the firmware or the programming. A supply chain compromise indicates that the attacker is able to compromise the manufacturing process itself to create vulnerabilities or any backdoor. It is challenging to detect such a supply chain compromise, and it is necessary to maintain strict quality and security manipulation during all stages of the supply chain [9].

2.3. Programming Layer

The interaction between a PLC and operators is enabled by the programming layer. The programming layer provides the logic (which are used to execute control operations) to the device. There are a variety of languages supported by PLCs, which encompasses traditional languages, such as C, and popular languages, such as Ladder Logic, Structured Text, and Function Block Diagrams [23]. Ladder Logic (LL) is a graphical sequential programming language that provides an intuitive interface to controllers for those who may not be familiar with traditional programming languages. In addition to LL, Function Block Diagram (FBD) is another graphical programming language. FBD provides an intuitive way to program with blocks. Simple routines expressed as state machines are generally programmed using Structured Text (ST).

Programming languages in PLCs enable programmers to define access rights to each controller's variable. "External" entities can be assigned privileges to remotely access variables, namely, Read/Write, Read Only, or None. Each variable is assigned Read/Write privilege by default [24]. The PLC may receive a Read or Write request with a variable from an external device. In this case, the PLC will look up the external request's variable to determine whether the Read or Write privilege should be granted. If so, the PLC will then return the value in a Read message or update the variable's value.

2.4. Firmware Layer

The programming layer and the hardware layer are connected together by the firmware layer and is referred to as the operating system, which encompasses lower-level functionalities, e.g., boot-loader code to initialize and load the operating system. It handles the basic behaviors of a device. The firmware controls the interactions (including physical inputs and outputs) between the operator and the device hardware. An attacker may attempt to gain access to the firmware on a PLC, thereby obtaining potentially unlimited control of the device. If the attacker is successful, the attacker has the capability to stealthily modify the device's behavior. The requirements of programming languages and system operations that needed to be complied by PLC products are defined in the IEC 61131-3 standard [23], which is followed by PLC vendors to program and compile the source code of control logic.

Traditionally, since it is the factory that installs the firmware and operators cannot reprogram devices, the PLC firmware is relatively secure against modification attacks. However, the firmware updating feature of PLCs to patch bugs and upgrade the firmware can be used by attackers to upload malicious firmware to the device. Checksum algorithms are commonly used to validate firmware updates before any installation and execution. A robust checksum algorithm plays an important role in preventing attackers from uploading malicious updates.

2.5. Scan Cycle and Control Logic

PLCs operate cyclically by repeating the same process, which is known as a scan cycle. Each scan cycle repeatedly execute the control logic program. In each scan cycle, network messages are uploaded from the network to local buffers and vice versa. In addition, the PLC reads information from sensors and locally stores their values, updates output signals from local values (to actuators), operates the control logic, and performs safety checks.

The control logic can be divided into several blocks: configuration blocks, data blocks, code blocks, and information blocks [25]. The configuration block covers other blocks' data such as a block's size and address, the PLC's IP address, and others. The data block contains variables (including inputs, outputs, timers, counters, etc., used in the code block) in the PLC. The Code block describes the compiled control logic code run by a PLC. When the control logic is retrieved from a PLC, information blocks will be used to recover it from the de-compiled source code.

2.6. PLC Vendors

Most large brands in industrial automation produce their own PLCs. These giant PLC manufacturers include Schneider Electric (Rueil-Malmaison, France), Mitsubishi Electric (Tokyo, Japan), Hitachi (Tokyo Japan), Siemens (Munich, Germany), Fuji Electric (Tokyo, Japan), and Panasonic (Osaka, Japan). There are also other PLC brand names such as Rockwell Automation (Milwaukee, WI, USA), Omron (Kyoto, Japan), Keyence (Osaka, Japan), Unitronics (Tel Aviv, Israel), Fatek (Taipei, Taiwan), Idec (Osaka, Japan), and Yokagawa (Tokyo, Japan) that have retained their focus on the production of PLCs and other industrial automation equipment.

2.7. Simulation Tools

Considering that vendors do not make the hardware and firmware information of their PLC products available, it is very difficult to conduct the research on PLCs' security.

To address such an issue, there have been efforts to produce open-source PLC technologies. The first open-source PLC was put forth by Souza [26] called MatPLC. This initial effort has several limitations. Firstly, it does not provide an interface with a programming integrated development environment (IDE). Secondly, it lacks a hardware platform to run the built-in physical Input/Output. Thirdly, it does not support the ladder logic. Tisserant, Bessard, and Sousa [27] developed an IDE for the IEC 61131–3 standards called PLCOpen Editor. Alves and Morris [28] developed an open-source PLC as OpenPLC by integrating the PLCOpen Editor with a compiler to build a comprehensive PLC package. The OpenPLC has an open source HMI editor, supports popular ICS protocols such as MODBUS [18] and DNP3 [19] and the open-source hardware.

There are several PLC simulation tools available. These can be divided into PLC System Simulators, PLC Core Simulators, and PLC Network Simulators [28]. PLC System Simulators (e.g., S7-PLCSIM [29], and RSLogix Emulate [30]) can simulate an entire PLC system, ranging from internal behaviors, the programming to network communications. PLC manufacturers usually provide PLC System Simulators. PLC Core Simulators (e.g., Common Open Research Emulator (CORE) [31], and AMICI [32]) normally simulate the PLC control logic program and the network behavior via the scripting language. PLC Network Simulators such as the Modbus Slave [33] focus on simulating the application layer of ICS network protocols. Thus, they are able to respond to network queries in the same way as regular PLCs, but they do not have any control over the built-in logic.

3. Adversarial Model

Depending on the goals of attackers, different assumptions have been made in the development of the threat model of PLCs.

3.1. Stealthiness

Some threat models assume that attackers target remaining stealthy from ICS operators such that the view of the HMI in the system does not indicate any effect caused by attacks. If an attack does not incur unintentionally observable effects, then it is a stealthy attack. For example, sensor readings should remain consistent with their expected values. To achieve such a goal, the attacker can compromise the HMI itself or launch attacks on the PLCs. Considering that the HMI is usually equipped with various security-enhancing techniques, the PLC becomes a more attractive target to attackers [34].

3.2. Control Logic

Some efforts focus on the symbolic execution for PLCs by assuming that attackers are able to update the control logic of the PLC [35]. Nevertheless, the modification of the PLC program is a challenging task in the real world. Usually, PLCs are armed with hardware-based mechanisms to protect them, which switch the PLC's status between "run" and "program" modes. The key of the PLC becomes unavailable when the PLC is in the "run" mode, and the PLC rejects any software modification. Therefore, unless the attacker is physically located in the plant and has access to the key, the attacker is unable to reprogram the PLC to change the PLC operation [36].

It is practical to assume that a remote attacker can break into the PLC network and forward the concrete information to the PLC, thereby changing the PLC's execution logic without physically modifying the PLC software.

3.3. Firmware

In terms of firmware attacks in PLCs, it is assumed that attackers have the access to the controller's program and can analyze the program [37]. Under this assumption, an attacker should be able to read any network message. Thus, attackers may attempt to read the internal variables from the Write variables (via network messages). However, if PLCs are equipped with hardware-based protections and there exist attestation methodologies [35], the attacker can neither update programs running in the PLC device nor modify PLC

signals to actuators [36]. In this case, the attacker's aim changes to utilize attack strategies to "force" a system's state to a critical state via an indirect operation of actuator states. For instance, the attacker may attempt to modify the controller's output variables (i.e., the signals sent to the actuators), and then change the physical state of the plant.

4. Different Types of Attacks

In this section, we first summarize existing control logic attacks on PLCs into control logic injection attacks and firmware modification attacks; then, we present an overview of network attacks on communication protocols running between PLCs (please note that we do not consider attacks such as behavioral anomalies on PLC-based systems (e.g., [38,39])).

4.1. Code Vulnerabilities

There might be "code errors", also known as code mistakes or flaws, in the logic implemented within the PLC program. These errors can lead to unintended behavior or malfunctioning of the control system. Following the summary in [40], several code-related vulnerabilities are summarized in Table 1. Code errors can be mitigated via debugging techniques.

Table 1. Code issues in PLCs.

Types	Descriptions
Logic Errors	Errors that could cause state transition, timing, control, and data flow issues.
Linkage and Scope Errors	Errors that handle the failure to or the deletion of the installation of a communication session between separate ladders.
Syntax Errors	Errors that were problematic in the compilation (not restricted). Such codes can be downloaded to the processor with at most one warning comparing to the individual downloading to the device.
Duplicate Objects	Objects such as timers and counters that have been defined more than once.
Unused Objects	Objects that were never used in the ladder logic but defined in the initial database which can be used for random functions.
Hidden jumpers	Software jumpers that avoid some parts of a rung in a ladder logic routine. They are not searchable and can be easily hidden from the untrained eye.

4.2. Control Logic Injection Attacks

Control logic injection attacks involve injecting malicious control logic into the PLC program to control the behavior of the industrial process or system being controlled. Such a type of attacks can modify the original control logic running on the PLC. Stunex [8] is one type of control logic injection attacks; it downloads the malicious control logic to the targeted PLC (i.e., Siemens S7-300) via the compromised engineering software. Langner [41] described how to inject rogue logic code into PLCs in 2011. McLaughlin presented [42] a malware that can automatically generate malicious payloads against a process control system and then designed SABOT [43] to automatically match control specifications in a PLC to adversarial instructions of the targeted control system's behavior, which enables an attacker to recover sufficient information of the PLC's internal structure, and thus, the attacker can compile and upload malicious payloads to the PLC to compromise the system. After a network scan, PLC-Blaster worm [44] can inject a malicious control logic to vulnerable PLCs (typically Siemens S7-1200), thereby causing the crash of the HMI software once the operator attempts to retrieve information from an infected PLC. Ghost in the PLC [45] demonstrated a PLC rootkit which breaks of the availability and integrity of an embedded system by exploiting certain Input/Output pin control operations to. Given that the PLC control logic could be interfered with normal engineering operations by the

attacker, Senthivel et al. [46] launched the denial of engineering operation (DEO) attacks against PLCs such that the software cannot work but the PLC continues execution. Yoo and Ahmed [25] achieved stealthiness in manipulating the control logic packet without making changes to the PLC firmware, which yielding the obfuscation functionality via data execution and fragmentation and noise padding control logic injection attacks. In 2022, two vulnerabilities were discovered in Rockwell Automation's PLCs that allow attackers to run malicious code on a PLC without triggering any obviously unusual behavior [47].

False data injection attacks [48] can be directly launched against PLCs allowing the attacker to learn partial information about the targeted subsystem and creating malicious results. However, if attackers do not take the operators' control commands into consideration when launching false data injection attacks, the forged system state may fail to meet the operators' expectations, and the attacks can be easily detected.

4.3. Firmware Modification Attacks

Firmware modification attacks involve unauthorized alterations to the firmware or software running on the PLC device itself, and they infect a PLC at the firmware level [25]. Beresford [49] explained how credentials can be extracted from remote memory dumps and how to power on and off PLCs via replay attacks against communication protocols running in Siemens S7 series PLCs. Meixell and Forner [50] released different methods to exploit PLCs by removing safety checks from the logic code. Basnight et al. [9] investigated the firmware update validation method and created a fake firmware sample to be uploaded and executed on a PLC that can cause an insecure checksum validation during the update process. Klick et al. [51] explored whether attackers can make use of exposed PLCs to extend their access to more PLCs. The rootkit HARVEY [34] is able to replace benign control commands with malicious commands to cause large-scale failures of the system. Bytes and Zhou [2] analyzed the software internals of WAGO PFC200 Series PLCs, and presented several potentially practical methods for the attack payload persistence executing on the firmware components. It has been notified in February 2022 that an unauthenticated and remote attacker has successfully launched denial-of-service (DoS) attacks to several Siemens PLCs [52]. In January 2023, a critical memory security detour vulnerability was identified within Siemens PLCs (SIMATIC S7-1200 and S7-1500) [53] that could disable access protection and give an attacker the ability to remotely execute malicious (or read and write) code everywhere on such PLC.

4.4. Attacks against Communication Protocols

Gao et al. [54] identified a series of data and command injection attacks and denial of service attacks in communication protocols applied by the PLC network that are caused by the lack of authentication. Fovino et al. [55] found several vulnerabilities in the MODBUS protocol due to its lack of security-enhancing techniques. Morris and Gao [56] described several attacks on MODBUS, including replay attacks, response and command injections, and denial of service attacks. Rahman et al. [57] launched denial of service attacks against MODBUS/TCP (which is a modified protocol of MODBUS applied over TCP/IP networks). Polge, Robert, and Traon [58] highlighted the impact of message flooding and eavesdropping attacks on an OPCUA application. Mathur and Tippenhauer [59] were successful in running a Man-In-The-Middle (MITM) attack between two PLCs capturing the data information and commands and re-writing them in addition to manipulating remote firmware and logic updates to each PLC. Wardak, Zhioua, and Almulhem [11] showed how password-based mechanisms can be compromised in a realistic scenario in recent versions of PLCs (since 2016). Cheng, Li, and Ma [60] demonstrated the vulnerabilities in the S7CommPlus protocols adopted by Siemens PLCs via the reverse debugging technique. Ghaleb, Zhioua, and Almulhem [12] launched three network attacks that could interfere with PLC communications and send arbitrary commands to the PLC. Ylmaz et al. [61] studied denial of service (DoS) attacks on the PLC and found that the delayed network traffic could lead to a slowdown in PLC operations and disable control over the PLC

control software. Robles-Durazno et al. [62] targeted network attacks against the PLC memory to interfere with system operations. Sandaruwan, Ranaweera, and Oleshchuk [10] analyzed several attacks, including brute-force attacks, replay attacks, MITM attacks, and authentication bypass attacks, which can be launched to affect the operations of PLCs.

4.5. Memory Attacks

While the majority of works focus on network traffic as a detection feature set [25], Cook, Marnerides, and Pezaros [63] were the first to propose a vendor-independent fingerprinting approach to detect memory attacks, which is a fingerprinting framework for PLC attack detection and classification. Based on the correlation between the dynamic and static behaviors in PLC registers (PLC registers are individual variables about concrete register areas (e.g., inputs, outputs, etc.)), they achieved the real-time composition of memory fingerprints in the PLC.

5. Countermeasures

In this section, we cover the existing efforts that mitigate threats and risks in the PLC network.

5.1. Firmware Integrity

Adelstein, Stillerman, and Kozen [64] introduced a detection method based on signature testing the integrity and execution flow using the detector when it was running. Symbolic execution (SE) [65] has been explored to identify whether there are attacks contained in PLC code. Canet et al. [66] proposed a framework for the automatic verification of PLC programs for conditional branches without implementing numerical instructions; however, this approach can cause state space explosion. McMinn and Butts [22] presented a verification tool for PLC firmware that can capture data during both the upload and download phases of the firmware without any modifications to the ICS system. Garcia [67] proposed an analysis method to perform static differential analysis of suspected changed PLC firmware via a variety of testing techniques comparing firmware models, firmware versions, and code differences. McLaughlin et al. [35] addressed attacks in which the malicious codes can be uploaded to the PLC through Trusted Safety Verifier (TSV) that combines SE and the model checking to verify whether the safety properties are satisfied by PLC programs.

5.2. Secure PLC Programs

The analysis on PLC binaries and source codes focuses on disassembly, i.e., decompilation, to find vulnerabilities and ensure the security of payload design, core codes, and general-purpose reconnaissance of controllers [68]. The PNF Software (refer to https://github.com/pnfsoftware) posted the closed-source JEB decompiler to run reverse engineering analyses on PLCs but only for Siemens S7 PLCs [69]. Keliris and Maniatakos [70] designed an open-source framework to enable binary analysis of general-purpose PLCs based on automated reverse engineering. Guo, Wu, and Wang [71] put forward SymPLC to automatically test whether PLC software written in programming languages are free of programming errors; however, this approach did not consider the timing parameter as a key component.

5.3. Defence Detection

Yao and Chow [72] introduced a methodology for defining detection rules to detect logic-changing attacks for PLC control logic that utilizes the control program logic change detector to detect the intentionally changed control logic. Abbasi et al. [73] effectively detected control flow hijacking attacks via a control flow integrity check tool, thereby guaranteeing the real-time availability of PLCs. Zonouz, Rrushi, and McLaughlin [74] put forward a way of detecting malicious code utilizing PLC code symbols. Feng et al. [75]

evaluated the security of PLCs through applying the fuzzy analytic hierarchy process, as well as the attack tree model.

5.4. Network Protocol Security

Communication protocols used by the PLC network are not designed with encryption, authentication, and authorization functions and, thus, cannot provide confidentiality, authentication, or integrity, and they are vulnerable to all kinds of network attacks. Majdalawieh, Parisi-Presicce, and Wijesekera[76] proposed a security framework for DNP3 by adding integrity, confidentiality, and authenticity to the original DNP3 where the encryption is applied to prevent eavesdropping attacks. Fovino et al. [55] enhanced the security of the MODBUS protocol by incorporating mechanisms to achieve integrity, authentication, non-repudiation, and anti-replay attacks. Sandaruwan, Ranaweera, and Oleshchuk [10] suggested several security measures to protect PLCs including authentication, timestamps, and intrusion detection systems. Voyiatzis, Katsigiannis, and Koubias [77] analyzed vulnerabilities in the MODBUS network protocol using the MODBUS/TCP Fuzzer to address the vulnerabilities in the MODBUS/TCP environment. Modbus.org published the MODBUS security protocol in 2018 to provide strong protection which is a combination of the Transport Layer Security (TLS) protocol and the traditional MODBUS protocol [78]. Malchow et al. [79] implemented a prototype called PLC Guard as a security solution for mitigating false network packets generated by communications between the PLC and other ICS components. Cheng, Li, and Ma [60] recommended solutions at the code level, design level, and protocol level to prevent replay attacks from the encryption algorithm applied by Siemens PLCs. Akpinar and Özçelik [80] modeled behaviors of field devices connected to the PLC and converted the model into ladder diagrams to mitigate PLC attacks. Zhang et al. [81] implemented a deep packet inspection (DPI) system to protect PLCs against from malicious network packet payloads.

5.5. Encryption

Heo et al. [82] demonstrated that the PLC communication network could be encrypted to achieve the data authenticity. Halas [83] revealed the need to identify encryption algorithms that are suitable for PLC devices. Zonouz, Rrushi, and McLaughlin [74] suggested the application of encyrption algorithms in PLC networks to reduce the possibility of reverse analysis on communication protocols.

To support the encryption running on PLCs, Alves, Morris, and Yoo [84] extended the open-source PLC OpenPLC [28] by adding AES-256 encryption and decryption capabilities. Later, Alves, Das, and Morris [85] modified the OpenPLC platform [28] to encrypt all data sent over the network [28].

6. Future Works

Security protection for PLCs should be considered for all ICSs. Existing research efforts on PLC security, which focus only on certain aspects of PLCs, cannot completely prevent vulnerabilities in PLCs. Key research directions for PLC security in the future can be summarized as follows.

6.1. Attacks

With the advancement of hacking skills, all components in ICSs need to be taken into consideration to evaluate the security of the PLCs. A simple study of one aspect–the PLC itself–cannot address all existing issues. Though some research papers may only address attacks on the PLC itself, many security incidents in ICSs target several vulnerabilities in different sectors of the ICS rather than PLC-only attacks. Therefore, it is necessary to conduct a systematic analysis of PLC attacks in combination with other components in ICSs. For example, future research can focus on attacking PLCs from the HMI or via the operation workstations in a stealthy way and corresponding solutions to prevent those attacks. In addition, there has been little attention paid to the security of PLC products such

as PLCnext [86]. It would be of interest to conduct a comprehensive security assessment of the PLCnext (or other small PLC brands) ecosystem, including the PLC hardware, firmware, software development environment, and communication protocols, to identify potential vulnerabilities and weaknesses.

6.2. Defense

In terms of defense, future research can emphasize the development of a protection framework for the comprehensive PLC network to prevent PLCs from being compromised with consideration to both the costs and functionalities. In addition, innovations can be conducted in the following areas.

- There are no common frameworks to conduct the formal validation of PLC code, and thus, a unified and effective approach for PLC code auditing is yet to be proposed.
- Considering that PLC programs are different from traditional computer programs, contributions to accurate detection and overhead mitigation are still expected.
- Encryption has been considered as an approach to enchance the security of communication protocols for PLC-based systems; however, applying encryption algorithms to the whole large-scale ICS network without affecting the normal operation of the service network is still a challenge.

7. Conclusions

In recent decades, several attacks have been successfully launched against PLC-based ICSs, which has caused significant damage to critical infrastructure. Since 2010, the security of PLCs has been seen as an increasing concern. Previous papers on PLC security either focus on attacks against PLCs or mitigation techniques to PLC vulnerabilities, but there have not been desirable efforts made to provide a comprehensive summary of PLC security. In this paper, we aim to bridge this gap by presenting an overview of PLC security based on the architectures and threat models of PLCs. We explored the fundamental components and communication protocols commonly used in PLC-based ICSs, highlighting the potential attack routes and vulnerabilities inherent in these systems. Finally, we concluded several future research areas in terms of PLC attacks and PLC defensive solutions.

Author Contributions: Conceptualization, H.C., J.H. and R.L.; methodology, H.C.; software, H.C. and J.H.; validation, H.C., J.H. and R.L.; formal analysis, H.C.; investigation, J.H.; resources, R.L.; data curation, H.C.; writing—original draft preparation, H.C.; writing—review and editing, J.H.; visualization, R.L.; supervision, J.H.; project administration, J.H.; funding acquisition, H.C., J.H. and R.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All data have been included in the article.

Acknowledgments: We would like to thank anonymous reviewers for commenting on earlier versions of this paper.

Conflicts of Interest: Rodney Louden is a lead engineer of company Critical Infrastructure Technologies. The authors declare no conflicts of interest.

References

1. Algburi, R.; Gao, H.; Al-Huda, Z. Design and implementation fuzzy-PLC temperature controller for the cooling tower to reduce dust emission in cement plant. In Proceedings of the World Scientific Proceedings Series on Computer Engineering and Information Science Developments of Artificial Intelligence Technologies in Computation and Robotics, WSPC, Cologne, Germany, 18–21 August 2020; pp. 1270–1279.

2. Bytes, A.; Zhou, J. Post-exploitation and Persistence Techniques Against Programmable Logic Controller. In *Lecture Notes in Computer Science, Proceedings of the Applied Cryptography and Network Security Workshops—ACNS 2020 Satellite Workshops, AIBlock, AIHWS, AIoTS, Cloud S&P, SCI, SecMT, and SiMLA, Rome, Italy, 19–22 October 2020*; Zhou, J., Conti, M., Ahmed, C.M., Au, M.H., Batina, L., Li, Z., Lin, J., Losiouk, E., Luo, B., Majumdar, S., et al., Eds.; Springer: Cham, Switzerland, 2020; Volume 12418, pp. 255–273. [CrossRef]
3. MarketsANDMarkets-Industrial Control Systems Security Market. Industrial Control Systems (ICS) Security Market by Component (Solution and Services), Solution, Security type (Network Security, Endpoint Security, Application Security, Database security), Vertical, and Region—Global Forecast to 2026. Available online: https://www.marketsandmarkets.com/Market-Reports/industrial-control-systems-security-ics-market-1273.html (accessed on 16 October 2023).
4. Chen, T.; Chen, S.; Tang, W.; Chen, B. Internet of Things: Development Intelligent Programmable IoT Controller for Emerging Industry Applications. *Sensors* **2022**, *22*, 5138. [CrossRef] [PubMed]
5. Gaspar, F.J.F.; González, I.; Calderón, A.J. Data acquisition and monitoring system framed in Industrial Internet of Things for PEM hydrogen generators. *Internet Things* **2023**, *22*, 100795. [CrossRef]
6. Smith, T. Hacker Jailed for Revenge Sewage Attacks. 2001. Available online: https://www.theregister.com/2001/10/31/hacker_jailed_for_revenge_sewage/ (accessed on 31 July 2023)
7. Johnson, R.E., III. Survey of SCADA security challenges and potential attack vectors. In Proceedings of the 5th International Conference for Internet Technology and Secured Transactions, ICITST 2010, London, UK, 8–10 November 2010; pp. 1–5.
8. Falliere, N.; Murchu, L.O.; Chien, E. W32.Stuxnet Dossier. 2010. Available online: https://www.wired.com/images_blogs/threatlevel/2010/11/w32_stuxnet_dossier.pdf (accessed on 12 September 2023).
9. Basnight, Z.; Butts, J.; Lopez, J., Jr.; Dubé, T. Firmware modification attacks on programmable logic controllers. *Int. J. Crit. Infrastruct. Prot.* **2013**, *6*, 76–84. [CrossRef]
10. Sandaruwan, G.P.H.; Ranaweera, P.S.; Oleshchuk, V.A. PLC security and critical infrastructure protection. In Proceedings of the 2013 IEEE 8th International Conference on Industrial and Information Systems, Peradeniya, Sri Lanka, 17–20 December 2013; pp. 81–85. [CrossRef]
11. Wardak, H.; Zhioua, S.; Almulhem, A. PLC access control: A security analysis. In Proceedings of the 2016 World Congress on Industrial Control Systems Security, WCICSS, London, UK, 12–14 December 2016; pp. 56–61. [CrossRef]
12. Ghaleb, A.; Zhioua, S.; Almulhem, A. On PLC network security. *Int. J. Crit. Infrastructure Prot.* **2018**, *22*, 62–69. [CrossRef]
13. Serhane, A.; Raad, M.; Raad, R.; Susilo, W. Programmable logic controllers based systems (PLC-BS): Vulnerabilities and threats. *SN Appl. Sci.* **2019**, *1*, 1. [CrossRef]
14. Wu, H.; Geng, Y.; Liu, K.; Liu, W. Research on Programmable Logic Controller Security. *IOP Conf. Ser. Mater. Sci. Eng.* **2019**, *569*, 042031. [CrossRef]
15. Pan, X.; Wang, Z.; Sun, Y. Review of PLC Security Issues in Industrial Control System. *J. Cyber Secur.* **2020**, *2*, 59–68. [CrossRef]
16. Telstar Inc. How SCADA, HMI, and PLC Work Together. 2019. Available online: https://www.telstarinc.com/how-scada-hmi-and-plc-work-together/ (accessed on 6 December 2023).
17. Institute of Electrical and Electronic Engineers. EtherNet/IP: Industrial Protocol White Paper. 2001. Available online: https://literature.rockwellautomation.com/idc/groups/literature/documents/wp/enet-wp001_-en-p.pdf (accessed on 5 October 2023).
18. Modbus Organization. Modbus. 1979. Available online: https://modbus.org/ (accessed on 15 August 2023).
19. DNP Users Group. DNP3. 1993. Available online: https://www.dnp.org/ (accessed on 1 December 2023).
20. Ait, R.; Yahia, A. PLC MQTT Communication Using TIA Portal, Mosquitto and Node-RED. 2023. Available online: https://www.solisplc.com/tutorials/plc-mqtt-communication-using-tia-portal-mosquitto-and-node-red#:~:text=MQTT%20is%20a%20protocol%20based,recipients%20subscribed%20to%20that%20topic (accessed on 5 December 2023).
21. PLC Table. PLC and OPC UA. 2023. Available online: https://www.plctable.com/plc-and-opc-ua/#:~:text=OPC%20UA%20can%20provide%20a,opportunities%20for%20optimization%20and%20development (accessed on 5 December 2023).
22. McMinn, L.; Butts, J. A Firmware Verification Tool for Programmable Logic Controllers. In *IFIP Advances in Information and Communication Technology, Proceedings of the Critical Infrastructure Protection VI—6th IFIP WG 11.10 International Conference, ICCIP 2012, Washington, DC, USA, 19–21 March 2012*; Revised Selected Papers; Butts, J., Shenoi, S., Eds.; Springer: Berlin/Heidelberg, Germany, 2012; Volume 390, pp. 59–69. [CrossRef]
23. PLCopen. International Electrotechnical Commission (IEC), Programmable Controllers—Part 3: Programming Languages. 2013. Available online: https://plcopen.org/iec-61131-3 (accessed on 9 November 2023).
24. Rockwell Automation. Logix 5000 Controllers General Instructions. 2020. Available online: https://literature.rockwellautomation.com/idc/groups/literature/documents/rm/1756-rm003_-en-p.pdf (accessed on 14 June 2023).
25. Yoo, H.; Ahmed, I. Control Logic Injection Attacks on Industrial Control Systems. In *IFIP Advances in Information and Communication Technology, Proceedings of the ICT Systems Security and Privacy Protection—34th IFIP TC 11 International Conference, SEC 2019, Lisbon, Portugal, 25–27 June 2019*; Dhillon, G., Karlsson, F., Hedström, K., Zúquete, A., Eds.; Springer: Cham, Switzerland, 2019; Volume 562, pp. 33–48. [CrossRef]
26. De Sousa, M. MatPLC-the truly open automation controller. In Proceedings of the IEEE 2002 28th Annual Conference of the Industrial Electronics Society, IECON 02, Seville, Spain, 5–8 November 2002; Volume 3, pp. 2278–2283. [CrossRef]

27. Tisserant, E.; Bessard, L.; de Sousa, M. An Open Source IEC 61131-3 Integrated Development Environment. In Proceedings of the 2007 5th IEEE International Conference on Industrial Informatics, Vienna, Austria, 23–27 June 2007; Volume 1, pp. 183–187. [CrossRef]
28. Alves, T.; Morris, T.H. OpenPLC: An IEC 61, 131-3 compliant open source industrial controller for cyber security research. *Comput. Secur.* **2018**, *78*, 364–379. [CrossRef]
29. Siemens. SIMATIC S7-PLCSIM—Software for SIMATIC Controllers. 2017. Available online: https://www.s7automation.com/tia-portal/ (accessed on 30 June 2023).
30. Rockwell Automation. Studio 5000 Logix Emulate. 2017. Available online: https://www.rockwellautomation.com/rockwellsoftware/products/studio5000-logix-emulate.page (accessed on 27 July 2023).
31. U.S. Naval Research Lab. Common Open Research Emulator (CORE). 2017. Available online: https://www.nrl.navy.mil/itd/ncs/products/core (accessed on 26 July 2023).
32. Genge, B.; Siaterlis, C.; Hohenadel, M. AMICI: An Assessment Platform for Multi-domain Security Experimentation on Critical Infrastructures. In *Lecture Notes in Computer Science, Proceedings of the Critical Information Infrastructures Security—7th International Workshop, CRITIS 2012, Lillehammer, Norway, 17–18 September 2012*; Revised Selected Papers; Hämmerli, B.M., Svendsen, N.K., López, J., Eds.; Springer: Berlin/Heidelberg, Germany, 2012; Volume 7722, pp. 228–239. [CrossRef]
33. Modbus Tools. Modbus Slave. 2012. Available online: https://www.modbustools.com/modbus_slave.html (accessed on 21 June 2023).
34. Garcia, L.; Brasser, F.; Cintuglu, M.H.; Sadeghi, A.; Mohammed, O.A.; Zonouz, S.A. Hey, My Malware Knows Physics! Attacking PLCs with Physical Model Aware Rootkit. In Proceedings of the 24th Annual Network and Distributed System Security Symposium, NDSS 2017, San Diego, CA, USA, 26 February–1 March 2017; The Internet Society: Reston, VA, USA, 2017.
35. McLaughlin, S.E.; Zonouz, S.A.; Pohly, D.J.; McDaniel, P.D. A Trusted Safety Verifier for Process Controller Code. In Proceedings of the 21st Annual Network and Distributed System Security Symposium, NDSS 2014, San Diego, CA, USA, 23–26 February 2014; The Internet Society: Reston, VA, USA, 2014.
36. Castellanos, J.H.; Ochoa, M.; Cárdenas, A.A.; Arden, O.; Zhou, J. AttkFinder: Discovering Attack Vectors in PLC Programs using Information Flow Analysis. In Proceedings of the RAID '21: 24th International Symposium on Research in Attacks, Intrusions and Defenses, San Sebastian, Spain, 6–8 October 2021; ACM: New York, NY, USA, 2011; pp. 235–250. [CrossRef]
37. Assante, M.J.; Lee, R.M. The Industrial Control System Cyber Kill Chain. 2015. Available online: https://na-production.s3.amazonaws.com/documents/industrial-control-system-cyber-kill-chain-36297.pdf (accessed on 4 July 2023).
38. Yau, K.; Chow, K.; Yiu, S.; Chan, C. Detecting anomalous behavior of PLC using semi-supervised machine learning. In Proceedings of the 2017 IEEE Conference on Communications and Network Security, CNS 2017, Las Vegas, NV, USA, 9–11 October 2017; pp. 580–585. [CrossRef]
39. Boateng, E.A.; Bruce, J.W. Unsupervised Machine Learning Techniques for Detecting PLC Process Control Anomalies. *J. Cybersecur. Priv.* **2022**, *2*, 220–244. [CrossRef]
40. Valentine, S.E., Jr. PLC Code V C Code Vulnerabilities Thr abilities through SCAD ough SCADA Systems A Systems. Ph.D. Thesis, University of South Carolina, Columbia, SC, USA, 2013.
41. Langner. A Time Bomb with Fourteen Bytes. 2011. Available online: https://www.langner.com/2011/07/a-time-bomb-with-fourteen-bytes/ (accessed on 24 July 2023).
42. McLaughlin, S.E. On Dynamic Malware Payloads Aimed at Programmable Logic Controllers. In Proceedings of the 6th USENIX Workshop on Hot Topics in Security, HotSec'11, San Francisco, CA, USA, 9 August 2011; USENIX Association: Berkeley, CA, USA, 2011.
43. McLaughlin, S.E.; McDaniel, P.D. SABOT: Specification-based payload generation for programmable logic controllers. In Proceedings of the the ACM Conference on Computer and Communications Security, CCS'12, Raleigh, NC, USA, 16–18 October 2012; ACM: New York, NY, USA, 2012; pp. 439–449. [CrossRef]
44. Spenneberg, R.; Brüggemann, M.; Schwartke, H. PLC-Blaster: A Worm Living Solely in the PLC. 2015. Available online: https://www.blackhat.com/docs/asia-16/materials/asia-16-Spenneberg-PLC-Blaster-A-Worm-Living-Solely-In-The-PLC-wp.pdf (accessed on 30 June 2023).
45. Abbasi, A.; Hashemi, M. Ghost in the PLC Designing an Undetectable Programmable Logic Controller Rootkit via Pin Control Attack. 2016. Available online: https://www.blackhat.com/docs/eu-16/materials/eu-16-Abbasi-Ghost-In-The-PLC-Designing-An-Undetectable-Programmable-Logic-Controller-Rootkit-wp.pdf (accessed on 2 August 2023).
46. Senthivel, S.; Dhungana, S.; Yoo, H.; Ahmed, I.; Roussev, V. Denial of Engineering Operations Attacks in Industrial Control Systems. In Proceedings of the Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy, CODASPY 2018, Tempe, AZ, USA, 19–21 March 2018; ACM: New York, NY, USA, 2018; pp. 319–329. [CrossRef]
47. Claroty Team82. Vulnerabilities in Rockwell Automation PLCs Could Enable Stuxnet-like Attacks. 2022. Available online: https://nvd.nist.gov/vuln/detail/cve-2022-1161 (accessed on 8 September 2023).
48. McLaughlin, S.E.; Zonouz, S.A. Controller-aware false data injection against programmable logic controllers. In Proceedings of the 2014 IEEE International Conference on Smart Grid Communications, SmartGridComm 2014, Venice, Italy, 3–6 November 2014; pp. 848–853. [CrossRef]
49. Dillon Beresford. Exploiting Siemens Simatic S7 PLCs. 2011. Available online: https://paper.bobylive.com/Meeting_Papers/BlackHat/USA-2011/BH_US11_Beresford_S7_PLCs_WP.pdf (accessed on 5 December 2023).

50. Meixell, B.; Forner, E. Out of Control: Demonstrating SCADA Device Exploitation. 2013. Available online: https://infocondb.org/con/black-hat/black-hat-usa-2013/out-of-control-demonstrating-scada-device-exploitation (accessed on 13 September 2023).
51. Klick, J.; Lau, S.; Marzin, D.; Malchow, J.; Roth, V. Internet-Facing PLCs—A New Back Orifice. 2015. Available online: https://www.blackhat.com/docs/us-15/materials/us-15-Klick-Internet-Facing-PLCs-A-New-Back-Orifice-wp.pdf (accessed on 19 Augustr 2023).
52. Kovacs, E. New Vulnerabilities Can Allow Hackers to Remotely Crash Siemens PLCs. 2022. Available online: https://www.securityweek.com/new-vulnerabilities-can-allow-hackers-remotely-crash-siemens-plcs (accessed on 22 October 2023).
53. Cox, C. EXPLOITED: Siemens PLCs, SIMATIC S7-1200 & S7-1500. 2023. Available online: https://embeddedcomputing.com/technology/security/exploited-siemens-plcs-simatic-s7-1200-s7-1500 (accessed on 6 December 2023).
54. Gao, W.; Morris, T.H.; Reaves, B.; Richey, D. On SCADA control system command and response injection and intrusion detection. In Proceedings of the 2010 eCrime Researchers Summit, eCrime 2010, Dallas, TX, USA, 18–20 October 2010; pp. 1–9. [CrossRef]
55. Fovino, I.N.; Carcano, A.; Masera, M.; Trombetta, A. Design and Implementation of a Secure Modbus Protocol. In I*FIP Advances in Information and Communication Technology, Proceedings of the Critical Infrastructure Protection III—Third Annual IFIP WG 11.10 International Conference on Critical Infrastructure Protection, Hanover, NH, USA, 23–25 March 2009*; Revised Selected Papers; Palmer, C.C., Shenoi, S., Eds.; Springer: Berlin/Heidelberg, Germany, 2009; Volume 311, pp. 83–96. [CrossRef]
56. Morris, T.H.; Gao, W. Industrial Control System Cyber Attacks. In *Workshops in Computing, Proceedings of the 1st International Symposium for ICS & SCADA Cyber Security Research 2013, ICS-CSR 2013, Leicester, UK, 16–17 September 2013*; Janicke, H., Jones, K.I., Eds.; BCS: Swindon, UK, 2013.
57. Rahman, A.; Mustafa, G.; Khan, A.Q.; Abid, M.; Durad, M.H. Launch of denial of service attacks on the modbus/TCP protocol and development of its protection mechanisms. *Int. J. Crit. Infrastruct. Prot.* **2022**, *39*, 100568. [CrossRef]
58. Polge, J.; Robert, J.; Traon, Y.L. Assessing the impact of attacks on OPC-UA applications in the Industry 4.0 era. In Proceedings of the 16th IEEE Annual Consumer Communications & Networking Conference, CCNC 2019, Las Vegas, NV, USA, 11–14 January 2019; pp. 1–6. [CrossRef]
59. Mathur, A.P.; Tippenhauer, N.O. SWaT: A water treatment testbed for research and training on ICS security. In Proceedings of the 2016 International Workshop on Cyber-physical Systems for Smart Water Networks, CySWater@CPSWeek 2016, Vienna, Austria, 11 April 2016; IEEE Computer Society: Los Alamitos, CA, USA, 2016; pp. 31–36. [CrossRef]
60. Cheng, L.; Li, D.; Ma, L. The Spear to Break the Security Wall of S7CommPlus. 2017. Available online: https://www.blackhat.com/docs/eu-17/materials/eu-17-Lei-The-Spear-To-Break%20-The-Security-Wall-Of-S7CommPlus-wp.pdf (accessed on 5 December 2023).
61. Ylmaz, 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; pp. 81–85. [CrossRef]
62. Robles-Durazno, A.; Moradpoor, N.; McWhinnie, J.; Russell, G.; Maneru-Marin, I. PLC memory attack detection and response in a clean water supply system. *Int. J. Crit. Infrastructure Prot.* **2019**, *26*. [CrossRef]
63. Cook, M.M.; Marnerides, A.K.; Pezaros, D. PLCPrint: Fingerprinting Memory Attacks in Programmable Logic Controllers. *IEEE Trans. Inf. Forensics Secur.* **2023**, *18*, 3376–3387. [CrossRef]
64. Adelstein, F.; Stillerman, M.; Kozen, D. Malicious Code Detection for Open Firmware. In Proceedings of the 18th Annual Computer Security Applications Conference (ACSAC 2002), Las Vegas, NV, USA, 9–13 December 2002; IEEE Computer Society: Los Alamitos, CA, USA, 2002; pp. 403–412. [CrossRef]
65. Schwartz, E.J.; Avgerinos, T.; Brumley, D. All You Ever Wanted to Know about Dynamic Taint Analysis and Forward Symbolic Execution (but Might Have Been Afraid to Ask). In Proceedings of the 31st IEEE Symposium on Security and Privacy, S&P 2010, Berleley/Oakland, CA, USA, 16–19 May 2010; IEEE Computer Society: Los Alamitos, CA, USA, 2010; pp. 317–331. [CrossRef]
66. Canet, G.; Couffin, S.; Lesage, J.; Petit, A.; Schnoebelen, P. Towards the automatic verification of PLC programs written in Instruction List. In Proceedings of the Proceedings of the IEEE International Conference on Systems, Man & Cybernetics: "Cybernetics Evolving to Systems, Humans, Organizations, and their Complex Interactions", Sheraton Music City Hotel, Nashville, TN, USA, 8–11 October 2000; pp. 2449–2454. [CrossRef]
67. Garcia, A.M. Firmware Modification Analysis in Programmable Logic Controllers. Ph.D. Thesis, Air Force Institute of Technology, Dayton, OH, USA, 2014.
68. Younis, M.B.; Frey, G. UML-based Approach for the Re-Engineering of PLC Programs. In Proceedings of the IECON 2006—32nd Annual Conference on IEEE Industrial Electronics, Paris, France, 6–10 November 2006; pp. 3691–3696. [CrossRef]
69. PNF Software. JEB. 2015. Available online: https://www.pnfsoftware.com/ (accessed on 28 July 2023).
70. Keliris, A.; Maniatakos, M. ICSREF: A Framework for Automated Reverse Engineering of Industrial Control Systems Binaries. In Proceedings of the 26th Annual Network and Distributed System Security Symposium, NDSS 2019, San Diego, CA, USA, 24–27 February 2019; The Internet Society: Reston, VA, USA, 2019.
71. Guo, S.; Wu, M.; Wang, C. Symbolic execution of programmable logic controller code. In Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering, ESEC/FSE 2017, Paderborn, Germany, 4–8 September 2017; ACM: New York, NY, USA, 2017; pp. 326–336. [CrossRef]

72. Yau, K.; Chow, K. PLC Forensics Based on Control Program Logic Change Detection. *J. Digit. Forensics Secur. Law* **2015**, *10*, 59–68. [CrossRef]
73. Abbasi, A.; Holz, T.; Zambon, E.; Etalle, S. ECFI: Asynchronous Control Flow Integrity for Programmable Logic Controllers. In Proceedings of the Proceedings of the 33rd Annual Computer Security Applications Conference, Orlando, FL, USA, 4–8 December 2017; ACM: New York, NY, USA, 2017; pp. 437–448. [CrossRef]
74. Zonouz, S.A.; Rrushi, J.L.; McLaughlin, S.E. Detecting Industrial Control Malware Using Automated PLC Code Analytics. *IEEE Secur. Priv.* **2014**, *12*, 40–47. [CrossRef]
75. Feng, T.; Shi, Y.; Gong, R.; Zhao, Q. The Security Assessment on Programmable Logic Controller based on Attack Tree Model and FAHP. In Proceedings of the 2019 IEEE International Conference on Power, Intelligent Computing and Systems (ICPICS), Shenyang, China, 12–14 July 2019; pp. 318–323. [CrossRef]
76. Majdalawieh, M.; Parisi-Presicce, F.; Wijesekera, D. DNPSec: Distributed Network Protocol Version 3 (DNP3) Security Framework. In *Proceedings of the Advances in Computer, Information, and Systems Sciences, and Engineering*; Elleithy, K., Sobh, T., Mahmood, A., Iskander, M., Karim, M., Eds.; Springer: Dordrecht, The Netherlands, 2006; pp. 227–234.
77. Voyiatzis, A.G.; Katsigiannis, K.; Koubias, S.A. A Modbus/TCP Fuzzer for testing internetworked industrial systems. In Proceedings of the 20th IEEE Conference on Emerging Technologies & Factory Automation, ETFA 2015, Luxembourg, 8–11 September 2015; pp. 1–6. [CrossRef]
78. Desruisseaux, D. Modbus Security—New Protocol to Improve Control System Security. 2018. Available online: https://blog.se.com/industry/machine-and-process-management/2018/08/30/modbus-security-new-protocol-to-improve-control-system-security/ (accessed on 5 December 2023).
79. Malchow, J.; Marzin, D.; Klick, J.; Kovacs, R.; Roth, V. PLC Guard: A practical defense against attacks on cyber-physical systems. In Proceedings of the 2015 IEEE Conference on Communications and Network Security, CNS 2015, Florence, Italy, 28–30 September 2015; pp. 326–334. [CrossRef]
80. Akpinar, K.O.; Özçelik, I. Analysis of Machine Learning Methods in EtherCAT-Based Anomaly Detection. *IEEE Access* **2019**, *7*, 184365–184374. [CrossRef]
81. Zhang, W.; Jiao, Y.; Wu, D.; Srinivasa, S.; De, A.; Ghosh, S.; Liu, P. Armor PLC: A Platform for Cyber Security Threats Assessments for PLCs. *Procedia Manuf.* **2019**, *39*, 270–278. [CrossRef]
82. Heo, J.; Hong, C.S.; Ju, S.H.; Lim, Y.H.; Lee, B.S.; Hyun, D.H. A Security Mechanism for Automation Control in PLC-based Networks. In Proceedings of the 2007 IEEE International Symposium on Power Line Communications and Its Applications, Pisa, Italy, 26–28 March 2007; pp. 466–470. [CrossRef]
83. Halas, M.; Bestak, I.; Orgon, M.; Kovac, A. Performance measurement of encryption algorithms and their effect on real running in PLC networks. In Proceedings of the 35th International Conference on Telecommunications and Signal Processing, TSP 2012, Prague, Czech Republic, 3–4 July 2012; pp. 161–164. [CrossRef]
84. Alves, T.; Morris, T.H.; Yoo, S. Securing SCADA Applications Using OpenPLC With End-To-End Encryption. In Proceedings of the Proceedings of the 3rd Annual Industrial Control System Security Workshop, ICSS 2017, San Juan, PR, USA, 5 December 2017; ACM: New York, NY, USA, 2017; pp. 1–6. [CrossRef]
85. Alves, T.; Das, R.; Morris, T.H. Embedding Encryption and Machine Learning Intrusion Prevention Systems on Programmable Logic Controllers. *IEEE Embed. Syst. Lett.* **2018**, *10*, 99–102. [CrossRef]
86. Phoenix Contact. Industries and Applications. 2023. Available online: https://www.phoenixcontact.com/en-au/ (accessed on 7 December 2023).