processes

MDPI

Review

A Survey on Programmable Logic Controller Vulnerabilities, Attacks, Detections, and Forensics

Zibo Wang [1,2], Yaofang Zhang [1,2], Yilu Chen [1], Hongri Liu [1,3], Bailing Wang [1,2,*] and Chonghua Wang [4,*]

[1] School of Computer Science and Technology, Harbin Institute of Technology, Weihai 264209, China
[2] School of Cyberspace Science, Harbin Institute of Technology, Harbin 150001, China
[3] Weihai Cyberguard Technologies Co., Ltd., Weihai 264209, China
[4] China Industrial Control Systems Cyber Emergency Response Team, Beijing 100040, China
* Correspondence: wbl@hit.edu.cn (B.W.); wangchonghua@cics-cert.org.cn (C.W.)

Abstract: Programmable Logic Controllers (PLCs), as specialized task-oriented embedded field devices, play a vital role in current industrial control systems (ICSs), which are composed of critical infrastructure. In order to meet increasing demands on cost-effectiveness while improving production efficiency, commercial-off-the-shelf software and hardware, and external networks such as the Internet, are integrated into the PLC-based control systems. However, it also provides opportunities for adversaries to launch malicious, targeted, and sophisticated cyberattacks. To that end, there is an urgent need to summarize ongoing work in PLC-based control systems on vulnerabilities, attacks, and security detection schemes for researchers and practitioners. Although surveys on similar topics exist, they are less involved in three key aspects, as follows: First and foremost, previous work focused more on system-level vulnerability analysis than PLC itself. Subsequently, it was not clear whether their work applied to the current systems or future ones, especially for security detection schemes. Finally, the prior surveys lacked a digital forensic research review of PLC-based control systems, which was significant for security analysis at different stages. As a result, we highlight vulnerability analysis at both a core component level and a system level, as well as attack models against availability, integrity, and confidentiality. Meanwhile, reviews of security detection schemes and digital forensic research for the current PLC-based systems are provided. Finally, we discuss future work for the next-generation systems.

Keywords: PLC-based control systems; vulnerabilities; attacks; security detection schemes; digital forensic

Citation: Wang, Z.; Zhang, Y.; Chen, Y.; Liu, H.; Wang, B.; Wang, C. A Survey on Programmable Logic Controller Vulnerabilities, Attacks, Detections, and Forensics. *Processes* **2023**, *11*, 918. https://doi.org/10.3390/pr11030918

Academic Editors: Xin Peng, Linlin Li and Hao Luo

Received: 10 February 2023
Revised: 11 March 2023
Accepted: 15 March 2023
Published: 17 March 2023

1. Introduction

Until recently, improvements of the ICS could be categorized into two main aspects: on one hand, with integration into external networks, efficiency in production and management has been greatly enhanced; on the other hand, adopting commercial software and hardware has improved economic profiles. However, both of these aspects cause a large number of vulnerabilities and exposures to cyberattacks, which will lead to catastrophic consequences.

The past decade has witnessed a number of ICS security incidents, such as Stuxnet [1], BlackEnergy [2], WannaCry [3], Triton [4], and so on. Among them, Stuxnet was viewed as a milestone since the attackers could compromise a Programmable Logic Controller to launch a sophisticated cyberattack. Owing to its important role in controlling physical facilities, the security of PLC-based control systems is supposed to be emphasized.

No matter whether in academia or industry, work on the security of PLC-based control systems can be divided into two parts: one is to provide some detection schemes to improve security outside the PLC, such as intrusion detection and honeypots; the other is to design a security built-in PLC equipped with verified programs, trustworthy firmware, and a perfect communication protocol. The former is always for current systems or legacy ones;

in contrast, the latter may be prepared for newly constructed production lines. Taking into consideration that current systems account for the majority of such instances, and that many industrial enterprises cannot afford the cost of production interruption and equipment replacement, more efforts should be made to improve the security of the PLC-based control systems in their current stage.

Besides, the current PLC itself has vulnerabilities in program verification, firmware, I/O memory, etc. Backdoors in general-purpose software and hardware, as well as flaws in communication protocols, are also introduced into the PLC-based control systems. Unfortunately, when attackers have access to certain industrial components, the aforementioned vulnerabilities could be exploited to form cyberattacks, e.g., command injection attacks, firmware modification attacks, memory corruption attacks, replay attacks, and so on. The increasing attack surfaces have the characteristics of being progress-oriented, stealthy, malicious, and sophisticated.

Based on existing vulnerabilities and attacks, research on the security detection scheme for the PLC-based control system has involved the perspectives of core-component defense and system-level defense, respectively. Compared with traditional Information Technology (IT) security schemes, keeping a balance between special industrial requirements and overall security is something that deserves to be considered. Diverse security solutions, such as code checking, firmware detection, traffic auditing, critical-state monitoring, etc., are bound to meet the industrial requirements, including real-time responses, continuous processing, frequent interactions, and high availability.

Meanwhile, the countermeasures should span the whole security lifecycle in the PLC-based systems. Differing from the solution mentioned above, digital forensic investigation of undesirable events is essential to the discovery of compromised components, internal or external attackers, malicious motivations, and skilled attack methods [5]. Moreover, according to the results of forensic research, many suggestions can be abstracted to secure overall PLC-based control systems in the future, ranging from the core component level to the system level.

Although there are several literatures on the security of PLC-based control systems, they lack a comprehensive review on vulnerabilities, attacks, and security detection schemes, focusing on the PLC itself and its interactions with other components. We provide a survey that pays more attention to the security of current PLC-based control systems and contains four parts: vulnerability analysis, attack models, security detection schemes, and forensic research. For further research, suggestions are outlined to strengthen security in future PLC-based control systems.

The main contributions of this article are summarized as follows:

- Compared with existing literature surveys on PLC-based control systems, the comprehensiveness of our work is demonstrated, containing vulnerabilities, attacks, security detection schemes, forensic research, and suggestions for the future of these systems;
- We analyze the vulnerabilities from two perspectives: the PLC itself and its relevant control systems. For the PLC itself, we mainly take the program, memory, and firmware into consideration. For its relevant control systems, we focus on application software, communication protocols, and connected devices with PLC;
- We provide the existing attacks on the PLC-based control systems in three categories, including attacks against availability, integrity, and confidentiality;
- For current PLC-based control systems, we present security detection schemes that are classified as program detection, firmware detection, fingerprint-based detection, intrusion detection, and honeypot-based detection;
- We discuss methodology, challenges, and achievements in forensic research for PLC-based control systems. For future construction of PLC-based control systems, six recommendations are outlined, concerning secure embedded systems, secure communication protocols, virtualization, open-source industrial control units, cloud-based or fog-based computing, and moving target defense (MTD).

As shown in Figure 1, Section 2 describes the background of the PLC and its relevant control systems, respectively. Our work is compared with existing surveys in Section 3. Respectively, Sections 4 and 5 provide vulnerabilities and attacks. Security detection schemes for current PLC-based control systems are presented in Section 6. Section 7 is used to discuss forensic research. We list some techniques or methods for future PLC-based control systems in Section 8. Finally, conclusions are drawn in Section 9. The list of acronyms used in the paper is given in Table A2.

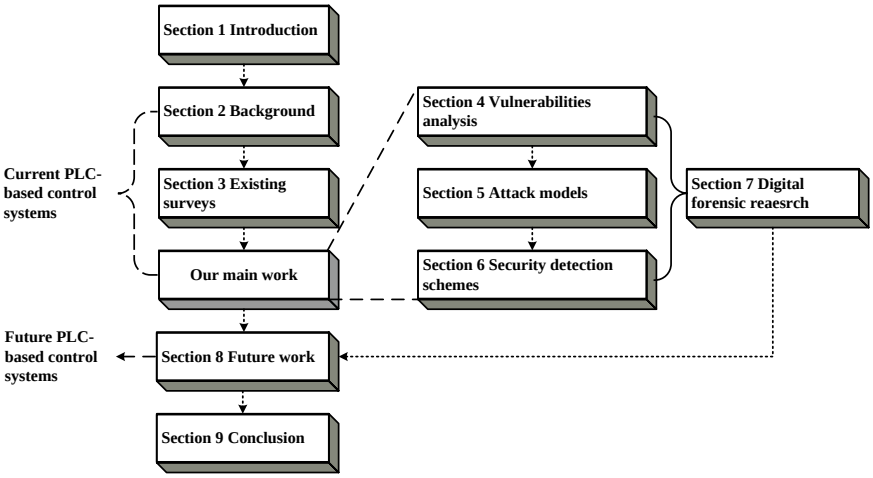

Figure 1. Review structure.

2. Background

Basic knowledge for the paper is provided in this section. Detailed information on PLCs and their relevant control systems is shown. Besides, requirements for ICS are considered by both attackers and defenders.

2.1. PLC Architecture

PLCs are a major component of control systems that interact with the real world. Functions of a PLC include logic, sequencing, timing, counting, arithmetic, and so on. The PLC conducts its input and output progress by receiving data from sensors and other field devices and translating it into digital values that can control devices in physics. PLC is mainly composed of programs, firmware, and hardware.

The PLC program directs the PLC to perform specific control logic tasks on the basis of input and output. It provides five programming languages: Ladder Diagram (LD), Structured Text (ST), Sequential Function Chart (SFC), Function Block Diagram (FBD) and Instruction List (IL). Among these languages, LD is the most commonly used one in practice [6]. Because it is similar to a relay circuit, it is easy to master for electrical personnel familiar with relay control. Firmware runs as the operating system (OS) in a PLC. Besides, reading and writing interactive data between the physical world and the PLC are also part of the firmware, which is regarded as the drivers for I/O hardware. Hardware typically consists of a CPU module, an input/output module, a programming device, memory, a communications interface, and a power module. Particularly, PLC has a small memory capacity, which contains two major parts: user memory and system memory. User programs store in the user memory, and some of the space is used by the system while the system memory is configured with the CPU. The CPU has access to this memory and an appropriate storage medium.

2.2. PLC-Based Control Systems

When it comes to PLC-based control systems, a supervisory control and data acquisition (SCADA) system is the most typical one. Hence, we first introduce the architecture of

a PLC-based control system, using the SCADA system as a prime example. Meanwhile, we summarize some communication protocols deployed by current PLCs.

In Figure 2, a modern SCADA system is made up of three layers: a supervisory control layer, an automatic control layer, and a physical layer. The supervisory control layer controls the monitoring operations of SCADA systems by gathering data. The automatic control layer is designed to regulate the operations of physical processes based on control commands. PLCs are located in this layer. The physical layer is where physical devices are running and is controlled by two layers above. A PLC controls individual actuators or sensors in the machinery. On the other side, connected devices such as a human–machine interface (HMI), engineering workstation, historian, and control server can communicate with the PLC. HMI obtains data from the PLC through a specific port, and it can read and write directly. Meanwhile, the engineering workstation is also uploading and downloading programs and configuration to the PLC.

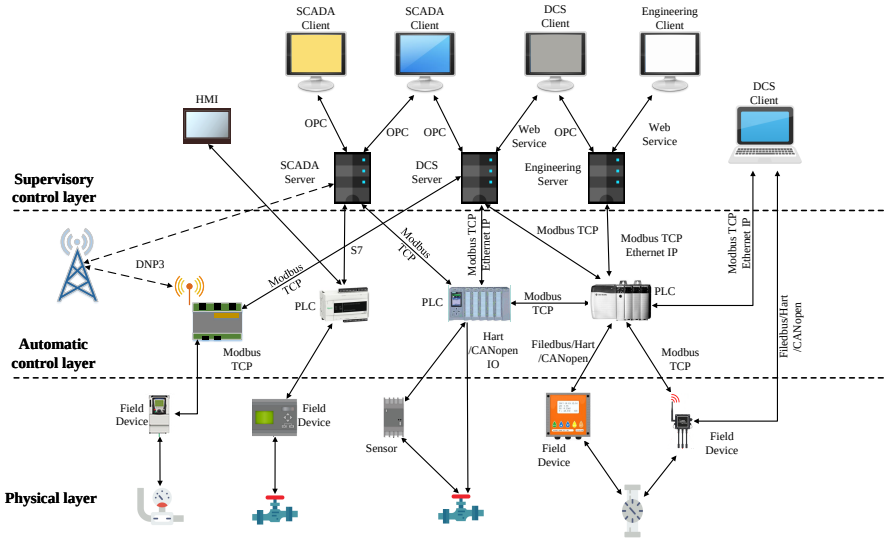

Figure 2. SCADA system architecture.

2.3. Requirements for ICS

Different from general IT system security, ICS has its own priorities for availability, integrity, and confidentiality. As a result, when we solve security problems for ICS, some special requirements should be considered as follows [7]: (1) high availability of each step in ICS; (2) integrity of industrial processes; (3) continuous operations along with a long operational life; (4) complex interactions with the physical world; (5) time-sensitive and real-time response; (6) distributed components; (7) multiple proprietary communication protocols; and (8) using large amounts of legacy subsystems.

3. Comparison with Existing Surveys

To collect relevant surveys, a series of search strings is provided, including "survey", "review", "PLC", "ICS", "vulnerability", "attack", "detection" and "forensics". We adopt such strings to manually search in the scopes of titles, abstracts, and keywords of existing papers. The papers are acquired from research databases such as IEEE Explore, ACM Digital Library, Science Direct, Web of Science, Springer Link, and MDPI. Additionally, we investigate the references of those located papers to find other relevant papers. Consequently, there were approximately 18 surveys published by mainstream journals and conferences on PLC-based control system security [7–24]. To some degree, none of the previous works provide a comprehensive analysis for PLC-based control systems, which is shown in Table A1.

3.1. Core Component Level vs. System Level

The ICS security is usually surveyed from two aspects: system level and component level. Due to the fact that PLC is a core component, its security issues are bound to attract more attention. However, most of them provided reviews for SCADA systems or the general ICS, while only four papers involved the security of the PLC itself [8,13,17,19]. Among the system-level review works, more efforts by industrial researchers were made on the network, especially for communication protocols [10,15,16].

3.2. A Thorough Overview of Vulnerabilities, Attacks and Security Detection Schemes

Although these surveys were more or less comprehensive with respect to vulnerabilities, attacks, and security detection schemes, they lacked a thorough overview of ICS security issues. For example, obvious discussions on vulnerabilities were not provided for PLCs and their relevant systems. Combined with the specific threats, the classification for attacks was not covered in detail. Furthermore, some security detection schemes were partly involved in a single kind [11,14,18].

3.3. Focus on Current PLC-Based Control System

Due to inadequate concern for security issues in the design stage, current legacy PLC-based control systems are vulnerable to sophisticated cyber threats. In Section 2, we introduced the fact that numerous legacy subsystems exist in practice. Nevertheless, the prior work did not take into full account the capability of the current systems to deal with certain cyberattacks, particularly in the analysis of security detection schemes.

3.4. The State of the Art in Forensic Researches

It is of the utmost importance to conduct a digital forensic investigation after a security breach, and this applies equally to ICS. For these 18 surveys, only one covered a little information on forensic science [12]. Ongoing forensic research for the core component and its relevant control systems was not presented.

3.5. Future Works for Future PLC-Based Control System

Even though it is a challenge to incorporate some recent security techniques into the current systems with little computational power, limited storage space, and low bandwidth, more advanced methodologies or strategies should be studied for future PLC-based control systems [21]. However, the aforementioned literature reviews hardly cover these aspects.

In a nutshell, our review work focuses on the open security issues and future direction for PLC-based control systems. We focus on the security of both the PLC itself and its relevant systems. A thorough analysis of vulnerabilities, attacks, and security detection schemes is presented. Compared with the similar surveys [7,12,22,24], we discuss the ongoing research regarding the achievements of digital forensic investigation as well as advanced security methodologies or strategies to give directions for securing ICS.

4. Vulnerability Analysis

In this section, we analyze vulnerabilities for the PLC itself and its relevant control system. Note that the whole list of vulnerabilities is collected from research papers. Figure 3 illustrates a classification between the component level and the system level. For current PLCs, the vulnerabilities mainly lie in programs, firmware, and memory. For control systems interacting with PLCs, vulnerabilities lie in industrial application software, communication industrial protocols, and connected devices.

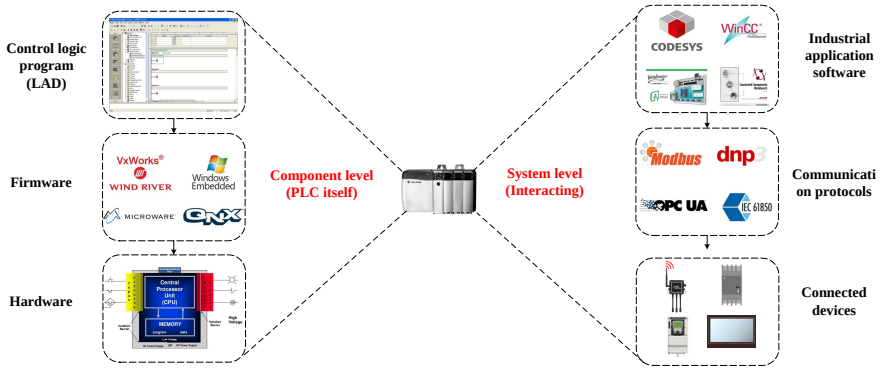

Figure 3. Component-level and System-level Classification.

4.1. Core Component-Level Analysis

It is well known that the design of PLCs did not fully take security into account. Given that PLCs are applied in all aspects of infrastructure, replacing legacy PLCs would be highly impracticable and almost impossible to implement. Both reasons tend to lead to various attacks targeting PLCs that exploit different vulnerabilities in their design.

4.1.1. PLC Program Vulnerabilities

Generally, control programs are written using the LD [6]. Considering the way that the programs are written or designed, there are some fatal flaws that exist in the programs. The integrity and availability of PLCs could possibly be violated, either intentionally or accidentally. From the perspective of programmers, back doors to hackers may be created, and PLC programs with dormant or unnoticed threats may be inherited; both are the result of a lack of professional knowledge and skills, such as using duplicated instructions, snooping, missing certain coils or outputs, bypassing, or denial of service [25,26]. When it comes to the ladder logic, it is vulnerable to malware insertion because of the lack of authentication before downloading programs into PLCs. One vitally important thing to emphasize is that the malware inserted into Ladder logic code could have a dormancy that can be broken down at any time, which poses a greater potential threat. For example, a malware in LD called Ladder Logic Bombs LLB was described by Govil et al. [27].

4.1.2. PLC Firmware Vulnerabilities

Actually, the firmware layer of PLCs is an embedded OS, e.g., Microware OS-9, VxWorks, and Microsoft Windows [8]. PLCs, which are admittedly complex embedded systems, can achieve their computing goals with the help of the OS. However, the firmware of the current PLCs is unsecure and vulnerable to attacks, just like the general OS. For example, the Beckhoff CX5020, which uses Windows CE 6.0 Plus, also has flaws that could be potentially attacked [28]. Furthermore, the number of the same kind of vulnerabilities in the typical microprocessor-based device was unexpectedly high. Attacking the PLC could be implemented without utilizing any flaws or vulnerabilities, simply by gaining access to the device and then exploiting its normal operation. These vulnerabilities may be exploited to execute firmware modification attacks and others that tend to affect the normal work of PLCs.

4.1.3. PLC Memory Vulnerabilities

As mentioned in Section 2, PLCs generally consist of the main memory, where the PLC stores the logic, and the register memory, which seems to be a temporary one for executing logic and is obtained by the processor in every scan cycle. Sandaruwan et al. [29] noted that some key variables that have influence on the main logic are contained in the register memory. Furthermore, the article showed that certain personal computers across the PLC network had access to the register memory and to read and write operations, which was

allowed by industrial plants. Consequently, an attack scenario is proposed: once an attacker injects any malicious values into the register memory after gaining access to any PLC, it seems that the system would suffer the threat of collapse. Exploiting the character, memory corruption attacks have the possibility of coming true.

4.2. System-Level Analysis

Besides programming, memory, and firmware, gaining the functions of PLCs requires combining them with industrial application software, communication protocols, and connected devices. The industrial application software can program, configure parameters, collect data, and monitor the status of PLCs. Communication protocols play a vitally important role in exchanging data among different devices. Also, the ports of connected devices have a direct association with the I/O of the PLC. However, they all still have negative potential influences on account of their own flaws.

4.2.1. Industrial Application Software Vulnerabilities

The application software consists mainly of PLC programming software, configuration software, and SCADA software. Once the software is compromised, the direct manipulation of PLCs is lost, which leads to the upload of malicious code, the modification of PLC parameters, the revealing of industrial data, and so on. Take Stuxnet for an indubitably representative example. As the first to contain rootkits specifically aimed at particular Siemens SCADA systems, Stuxnet exploited four zero-days, which was its uniqueness. In addition, Leverett and Wightman [30] illustrated that CoDeSys, a third-party programming software, had the capability to modify the code within PLCs, to maintain process control integrity, and to inspect directories for traversal, which possibly became the basis of malicious exploits and caused code injection attacks.

4.2.2. Communication Industrial Protocols' Vulnerabilities

Although the current network protocols work efficiently for communication between PLCs, the security of the protocols is not considered in their design. There are major flaws in the authentication, authorization, encryption, availability, integrity, and confidentiality of protocols. Three common vulnerabilities are summarized: (1) Lack of authentication: Malicious users can easily gain privileges and forge protocol packets without any identification mechanism. (2) Lack of authorization: Malicious users can abuse the function code in order to send any messages to others for useful feedback. (3) Lack of encryption: Malicious users can capture transparent data for achieving malevolent attempts. Take S7 protocol vulnerabilities for example. Hui and McLaughlin manifested that this protocol lacked authentication, which caused attacks on Siemens PLCs. In addition, such vulnerabilities could be utilized to implement replay attacks, MITM, and so on [31].

4.2.3. Connected Devices' Vulnerabilities

The integrity and reliability of PLC operation could be affected by the improper or fake status and input/output of connected devices. Serhane et al. [26] indicated that HMIs, data transfer units, and historian terminal units became more vulnerable due to increasing remote access, which drew attention from a number of attackers. In particular, these connected devices were regarded as having vulnerable initial access to infect networks. Therefore, PLC programs are also influenced by the wrong commands and input or output values from them, and even further errors occur.

5. Attack Models

We discuss attack models for the PLC-based control systems in this section. There are 15 types of attacks collected from existing research articles that focus on the PLC itself and its relevant systems. In our review, we classify the attacks into three categories, as shown in Figure 4, containing: (1) attacks against availability; (2) attacks against integrity; and (3) attacks against confidentiality. Our classification is based on the behavior of attacks

and corresponds with the vulnerabilities in the previous section. The main findings are featured in Table 1.

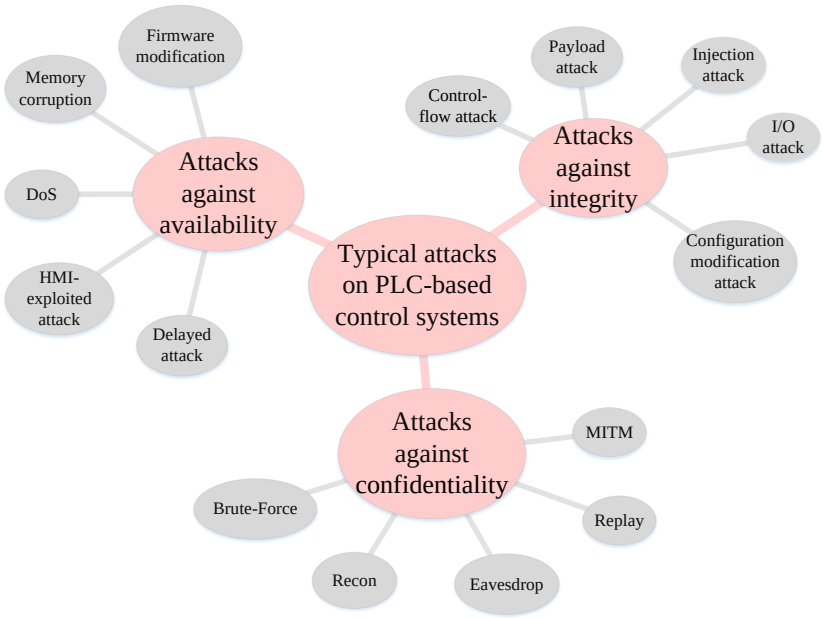

Figure 4. Classification of attacks in PLC-based control systems.

Table 1. Main findings in attack models.

Attack	Target	Ref.
Firmware Modification	PLC firmaware layer	[32,33]
Memory Corruption	PLC I/O Memory	[34,35]
DoS	Service or resource of control system	[36–39]
Payload Attack	PLC control logic	[40–42]
Injection	Messages with insecure protocols	[43–49]
I/O Pin Control	PLC pin controller	[50,51]
MITM	Messages with insecure protocols/ poor authentication	[29,52–56]
Replay	Messages with insecure protocols/ poor authentication	[29,52,54,57]

5.1. Attacks against Availability

We classify five attacks in this category: firmware modification attack, memory corruption attack, Denial of Service (DoS) attack, delayed attack, and HMI-exploited attack. From the latest works, we introduce three typically serious attacks.

5.1.1. Firmware Modification Attacks

Because of vulnerabilities within firmware update validation, PLCs are exposed to firmware modification attacks. Besides, the flaws of protocols, including their lack of source or data authentication, can be exploited to send commands to these unsafe components. Once they have attacked successfully, they can attack other components on the control system, which tends to control all inputs and outputs of PLCs, give wrong commands, and cause damage to the physical equipment.

Basnight et al. [32] indicated that the first step of firmware modification was usually to reverse a binary firmware, which could gain specific functions through disassembly, and that it was difficult to extract and verify a malicious firmware on running PLCs. As a result, PLC firmware modification became an intrusive and least detectable attack. Garcia et al. [33] constructed a PLC rootkit named HARVEY to attack against cyber-physical

power grid control systems. The model-aware rootkit infected the firmware of PLCs and tampered with all inputs and outputs of the PLCs arbitrarily.

5.1.2. Memory Corruption Attacks

There are several memory corruption vulnerabilities in current protocol implementations, such as buffer overflows and faulty mappings between protocol elements and main memory addresses. Consequently, after having access to a control network, attackers could corrupt control data, configuration data, and decision-making data. PLC memory corruption attacks could overwrite the memory locations relevant to I/O and tamper with the setpoint variables.

Focusing on attacking the PLC input memory, Robles-Durazno et al. [34,35] showed their attack methodology, which sent crafted packets to the input memory, assuming that attackers have access to the control network. They also proposed three PLC memory corruption attacks that targeted three areas of the memory. The input memory was attacked by overwriting the bytes of memory allocated to the outside sensors. In a similar way, the memory associated with the outputs was also affected. Attacking working memory aimed to modify setpoint variables like a high or low alarm setting value in control systems.

5.1.3. DoS Attacks

Current PLCs may respond to every read request arriving from any IP or MAC address without any filters. Therefore, one of the critical threats to ICS is a DoS attack. Technically, a DoS attack is a goal consisting of other attacks, not a potential attack type. DoS limits the availability of a service or resource in part or as a whole whose aim is to hinder legitimate access to authorized resources as well as disturb resource utilization in its intended way.

Tacliad et al. [36] discovered a type of DoS attack by using Ethernet/Industrial Protocol (ENIP) Fuzz to execute Programmable Controller Communication Command (PCCC) service, where an invalid data file type caused failures in reading data. In addition, a DoS attack is a direct IP-oriented attack. Ylmaz et al. [37] implemented DoS attacks by using the spoofing method with bogus IP addresses between PLCs and Total Integrated Automation (TIA) Portal. Similarly, Sayegh et al. [38] also exploited IP packet flooding to attack PLC. More specifically, they launched DoS attacks between PLCs and HMIs in four ways, respectively. In addition, Niedermaier et al. [39] demonstrated a DoS attack on PLC by using Generic Routing Encapsulation (GRE) based SYN packets with Netwox tool #76.

5.2. Attacks against Integrity

We classify five attacks in this category: payload attack, injection attack, I/O pin control attack, control-flow attack, and configuration modification attack. From the latest works, we introduce three typically serious attacks.

5.2.1. Payload Attacks

PLCs provide hardware peripherals and firmware support for control logic that is usually called “payload”. The increase in attention is being paid to the potential malicious programs uploaded directly to PLCs. Once proper privileges are obtained, warnings of payload changes could always be altered by attackers [40]. Thus, engineers cannot find out the malicious payload by performing integrity checks on specific PLC programming software from vendors in time. Accordingly, payload attacks could be treated as modifying the logic or uploading a malicious program directly to the PLC.

According to popular opinion, it is unpractical to attack automatically against PLC without a priori knowledge of the target physical process. Focusing on the ability of a program to generate a malicious payload, McLaughlin [41,42] analyzed problems in the design of PLC malware that observed the process and then generated a dynamic payload. They indicated that this design markedly lowered the bar for attacking PLCs. Subsequently, a tool called SABOT was proposed by McLaughlin and McDaniel. They made it possible to launch an attack with less prior knowledge of a target PLC-based control system in

an automatic way through a mapping between PLC instructions and specifications that described system behaviors by adversaries.

5.2.2. Injection Attacks

PLC-based control systems indiscriminately believe messages received over the network using insecure communication protocols. Moreover, PLC offers an open-source function library allowing the establishment of TCP/UDP communication, which poses a high risk. Therefore, attackers can have access to control systems and inject malicious data into PLCs, thus gaining control and causing the denial of engineering operations, which is known as an "injection attack".

A kind of injection attack named the "denial of engineering operation" (DEO) attack was introduced in three scenarios [43–45]. The goal of the DEO attack was to interfere with normal operations by engineers while updating control logic during the progress of a download or upload. In DEO attacks I and II, they both needed a middle position between an engineering workstation and the PLC. The former hid infected ladder logic from a program. And then the latter uploaded the malformed logic to crash the program software directly. While, a well-designed binary ladder logic program was developed to inject into the target PLC in the DEO attack III. When the software tried to acquire the program from the PLC, an exception was thrown to manifest a failure.

Another type of injection attack concerns false data. McLaughlin and Zonouz [46] presented a false-data injection attack with controller awareness for individual PLCs. The attack tool analyzed the I/O trace to construct an internal logic model of the compromised PLCs and then generated a series of inputs to meet expected PLC outputs. Xiao et al. [47] proposed a false sequence attack to build a discrete event model by using a collection of fault-free I/O traces. After that, a set of false sequences could be obtained from the model to inject into compromised sensors. Based on signal-interpreted Petri nets, a false data injection was modeled to manipulate the sensor measurements so as to change state variables in a stealthy manner [48]. Given stealthy methods to achieve injection attacks, Yoo and Ahmed [49] proposed a data execution attack and a fragmentation and noise padding attack, transferring the control logic to PLCs over the network.

5.2.3. I/O Pin Control Attack

As typical embedded equipment, the PLC combines multiple I/O interfaces, managed by a pin controller. Pin controller behavior is determined by a series of registers. Attackers can launch the I/O pin control attack to breach the integrity of legitimate operations, tampering with the interaction with the physical world.

For targeting PLC, assuming that the root access and the other necessarily prior knowledge had been gained, Abbasi [50] proposed an I/O manipulation attack in a stealthy manner that used the processor debug registers to intercept read or write operations, differing from typical function hooking techniques. The author stated that the manipulation was unknown to the PLC runtime software itself. Based on the previous research, Abbasi and Hashemi [51] demonstrated two different attack implementations, regarding reliable manipulation with root access as well as less reliable manipulation without root access.

5.3. Attacks against Confidentiality

We classify five attacks in this category: Man-In-The-Middle (MITM), replay, eavesdrop, bypass, and brute-force. Among them, MITM and replay are discussed.

5.3.1. MITM Attack

In the MITM attack, the attacker always places himself between PLCs and control centers. Due to the lack of encryption and authentication in current industrial protocols, both end points could have been expected to communicate with each other legitimately. What is worse, the attacker is prone to manipulating messages to violate confidentiality.

When it comes to the MITM attack, the most common method is Address Resolution Protocol (ARP) poisoning. It relates the IP address of victims to MAC addresses in the ARP tables of attackers. The attack scenario about MITM was demonstrated, which intercepted all packets between the engineering station and PLC using the Siemens S7COMM Protocol [29,52–54]. Lim et al. [55] deployed the MITM attack in the Tricon PLC system to induce common-mode failures. It intercepted and modified the configuration packets, subsequently redirecting the traffic through the compromised PLCs. Moreover, Grandgenett et al. [56] devised a MITM attack to filter and modify the CIP data and commands between RSLogix 5000 and the EtherNet/IP Web Server Module.

5.3.2. Replay Attack

A replay attack is treated as a way to exploit a system function by retransmitting a legitimate message. The message is usually included in packets that are captured at the source or intercepted via man-in-the-middle. Attackers aim to pass through the authentication process successfully while they do not completely grasp the protocol or mechanism of the PLC-based control systems.

At Black Hat USA 2011, Beresford [52] presented an attack on a Siemens S7 PLC by using a replay attack. His presentation inspired many researchers who were interested in PLC security [29,54]. Due to the flaws in the S7COMM protocol, S7COMM-Plus was developed with anti-replay protection. However, Lei et al. [57] discovered that the protocol still had the opportunity of being exploited, and then they cracked the encryption algorithms.

In a word, we classified the attacks mentioned in the ongoing research papers based on the three categories. In the aforementioned subsection, we mentioned the names of some attacks but did not give detailed descriptions because of the following two criteria: (1) The existing papers involving related attacks account for a relatively small proportion, such as delayed attack and HMI-exploited attack, and configuration modification attack; (2) attack skills are similar to those in the IT domain, such as control-flow attack, eavesdrop, recon, and brute-force. We further briefly summarize the descriptions of those attacks in Table 2. In addition, some complicated attacks were actually a combination of the above, such as PLCinject [58], PLC-Blaster [59], and PLC-PC worms [60]. To some degree, they focus more on the entry or propagation of such attacks. Nevertheless, it does not fall within the scope of our review.

Table 2. Brief summaries on the remaining attacks.

Attack	Tampering Category	Description	Ref.
Delayed attack	Availability	Packet delays on the network are purposely designed and injected into the control system to result in the loss of system stability.	[61]
HMI-exploited attack	Availability	A compromised HMI provides an opportunity to intercept the communication channel with PLCs and alter physical operating states.	[62]
Control-flow attack	Integrity	Manipulating the execution flow of a process results in the execution of arbitrary code.	[50]
Configuration modification attack	Integrity	Critical parameters are modified to force PLCs out of control.	[50]
Eavesdrop	Confidentiality	Read critical messages between two communication devices.	[63]
Recon	Confidentiality	Gather information precedes subsequent attacks such as addresses and function codes.	[5]
Brute-force	Confidentiality	Enumerate possible orders to crack the PLC authentication algorithm.	[64]

6. Security Detection Schemes

We discuss security detection schemes for PLC-based control systems in this section. Note that these schemes are all from the existing research articles. The classification is shown in Figure 5. For the PLC itself, we introduced the security detection schemes, including program detection, firmware detection, and fingerprinting-based detection. Subsequently, from a system-level aspect, we provide intrusion detection as well as honeypot-based detection. It should be noted for readers that the rough classification is mainly for organizational reasons.

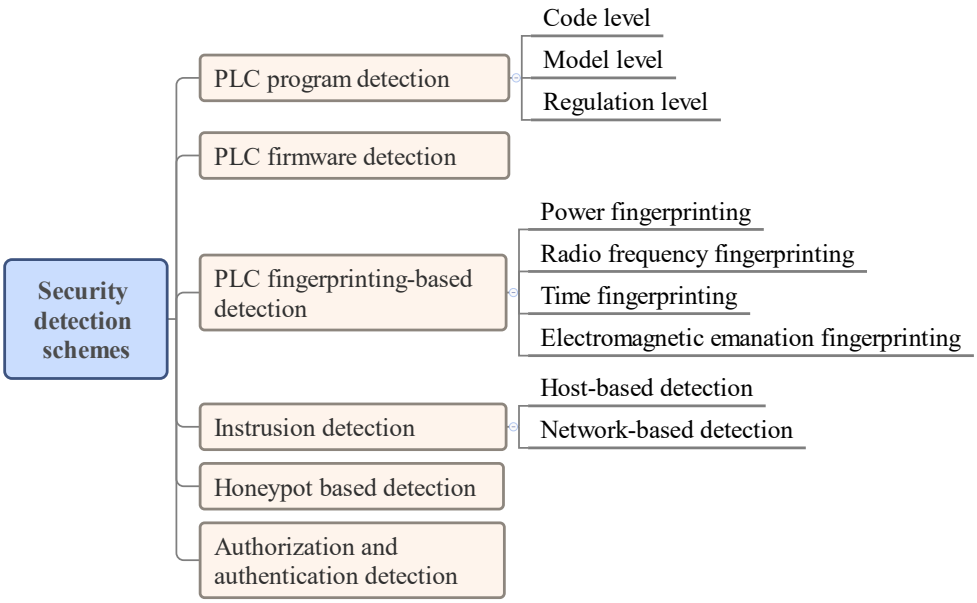

Figure 5. Classification of security detection schemes in PLC-based control systems.

6.1. PLC Program Detection

Running states of PLC affect working states of the whole system, whereas the normal running of PLC depends on the inerrant operation of its logical program. Meanwhile, the PLC program is vulnerable to attacks such as the payload attack and the injection attack, which seriously violate security requirements and even cause serious physical damage. Therefore, program detection is a vital part of the PLC's running state.

Initially, PLC program detection is more concerned with reliability, stability, and safety. As the ICS interconnected with the outside world via networks, the PLC program suffered from more network attacks, and the detection of its own security was given more attention by researchers. As a result, security problems are inseparable from safety issues. Therefore, an overview of program detection is demonstrated in this subsection. After that, we provided the detection methods for safety and security, respectively.

In general, our work is inspired by the hierarchy framework on compositional verification for PLC program detection stated in [65]. The framework introduces program detection in three aspects, including code, model, and statute. Code-level verification focuses on the coding analysis. Mode-level verification focuses on dynamic behaviors. Statue-level verification focuses on static properties. Regarding the verification methodologies at the three levels, we discussed the impact of program detection on safety and security for the PLC program in the following sections.

6.1.1. Program Detection on Safety

1. Code-level detection: With the help of an Aachen Rigorous Code Analysis and Debugging Environment (ARCADE). PLC verification platform, Stattelmann et al. [66] applied static code analysis to an industrial software development environment in engineering. Recently, Zhang et al. [67] presented a static program analysis approach,

named VETPLC, which built timed event causality graphs for causal relations among events in PLC code. Hence, it could be applied to automatically detect hidden safety violations;

2. Model-level detection: At this level, model checking is widely used in IT fields, where it constructs a formal model for the system and then explores the whole set of states in a brute-force manner in order to verify the property [68]. We focus on the application of the methods in specific PLC programming languages and the application of some model checking tools in PLC-based control systems in the next subsection;
3. Statue-level detection: For statue-level detection, theorem proof is applied to the verification of the correctness of PLC programs in each scanning cycle. Coq is a theorem-proving tool based on the calculus of inductive constructions and excellent mathematical models. Xiao et al. [69] defined the formal semantics of typical PLC programs with an extended λ-calculus definition and provided a Coq-based verification.

From a practical point of view, the following three aspects deserve equal attention for PLC safe program detection: (1) Modular code makes it easier to test and track the integrity of code modules. For example, if the code within a module has been thoroughly tested, any copies of those modules will be checked against the hash of the source code. Verification will be useful if the integrity of the code is in question after an incident; (2) PLCs are not in "RUN" mode, and then the code can be changed. Some PLCs have a checksum to notify code changes. However, some legacy PLCs lack such a safety mechanism. For example, an alarm signal should be configured to warn operators at the end of a shift; (3) During code development, engineers need to test and validate their software modules by substituting data outside of expected boundaries. Assign different locked memory segments for firmware, logic, and the protocol stack in order to detect abuse.

6.1.2. Program Detection on Security

In our review, we found that researchers began to shift their interest from safety to security for PLC program detection after the year 2014. Combined with the symbol execution method, model checking could be applied to detecting attacks with the malicious PLC code. McLaughlin et al. [70] proposed a trusted safety verifier, which is a trusted computing base for the verification of the PLC critical logic. The symbolic execution combined with the model checking algorithm provided a last-step verification of commands before reacting to actual controlled devices. In the same year, Zonouz et al. [71] presented a similar method to detect malicious code bound to PLCs that did not violate the security requirements of the underlying physical factory. A temporal execution graph was built with symbolic execution outputs for modeling the PLC's subsequent I/O.

Chang et al. [72] presented a detection method of malicious behaviors by the state verification, which considered the sequence of input vectors. On the other hand, they removed duplicate executable paths in the same scan cycle to tackle the state space explosion problem. Moreover, the aforementioned tools were applied by researchers for PLC program detection as well. Modeling with NuSMV, Kottler et al. [73] focused on verifying PLC programs in LD and ST languages to identify certain security vulnerabilities. Hailesellasie and Hasan [74] proposed a scheme based on differences in the attributed graph between potentially compromised PLC programs and trusted ones. The graphs were generated from UPPAAL-based formal modeling.

Model checking cannot handle binary code directly. As a result, the reverse engineering techniques for the binary PLC codes deserve to be researched. Lv et al. [75] proposed a decompiling framework that was suitable for the PLC program based on templates in the form of instructions or operands. Likewise, Keliris and Maniatakos [76] developed an industrial control systems reverse engineering framework for reversing PLC binaries compiled with CODESYS, taking into consideration their unique domain-specific characteristics. Chang et al. [77] disassembled the program into STL and then built a Control Flow Graph (CFG) to get a mapping between outputs and inputs. On the other hand, due to the fact that PLCs are vulnerable to control-flow hijacking attacks, the CFG, which is extracted

from the binaries, is bound to be of great importance. Abbasi et al. [78] introduced an embedded a Control Flow Integrity (CFI) mechanism named ECFI. In their work, they considered both real-time and runtime operations for real-world industrial PLCs.

6.2. PLC Firmware Detection

PLC firmware provides a link between the hardware and software. It has been proven that PLCs lack firmware auditing capabilities. Once the firmware is manipulated by attackers or malware, it can control other physical system components via the compromised PLCs. Therefore, firmware modification detection is extremely important for PLC-based control systems.

McMinn and Butts [79] designed a firmware verification tool for the serial data while uploading. The tool could be applied to a variety of platforms without requiring any modifications to the existing PLC-based control systems. Furthermore, an approach to inferring the firmware update validation was proposed by Basnight et al. [32] by using reverse engineering techniques. To detect stealthy firmware modifications like HARVEY, Garcia et al. [33] gave suggestions: (1) allowing to check the PLC firmware integrity; (2) monitoring data from sensors to PLCs; (3) monitoring data from PLCs to actuators.

6.3. PLC Side-Channel Detection

Research has shown much unintentional information leakage from PLCs, such as radio frequency (RF) emissions, power consumption, electromagnetic (EM) emanations, and operation time. Such leakage information is usually obtained by a series of physical measurements from a side channel. Thus, side-channel analysis is a common method to detect malicious attacks or unintended operations in PLCs. We discuss four techniques for side-channel detection.

6.3.1. Radio-Frequency-Based Detection

In 2012, Stone and Temple [80] proposed a RF-based methodology to detect anomalous operations of PLCs. After that, their research group improved the anomaly detection capability on the basis of the previous research by adopting Hilbert transformed sequences of unintentional time domain emissions of PLCs [81]. Moreover, an additional advantage was that the standalone RF-based analysis system only depended on physical layer information from PLCs, which was isolated from network-based cyberattacks.

6.3.2. Power Fingerprinting Detection

Leveraging the power of fingerprinting, Gonzalez and Hinton [82] presented a method to monitor PLCs and detect malicious software execution. However, it is not feasible for real-time monitoring because power fingerprinting was collected from a sensor that closely interacted with the CPU of PLCs. In other words, the exposure of CPUs to sensors and high-frequency data acquisition means burdens for PLCs. To overcome these limitations, Xiao et al. [83] demonstrated a real-time detection method in a non-invasive way, relying on a resistor to collect power consumption traces.

6.3.3. Time-Based Detection

The above-mentioned techniques depended on the additional hardware to be added to PLC-based control systems, which made the approach complex. Using timing-based side channels, Dunlap et al. [84] presented an approach to detect unauthorized modifications of PLCs by execution time measurements.

6.3.4. EM Emanation-Based Detection

Boggs et al. [85] demonstrated the feasibility of EM emanation-based detection for code execution on PLC-based control systems. They separately monitored normal and abnormal activities via a signal cliff detection method. Likewise, Van Aubel et al. [86] leveraged EM side-channel measurements for detecting behavior changes in an executing

industrial software. Particularly, they suggested two layers of verification. The first layer checked the user program runtime, while its EM trace was compared with a baseline version in the second layer.

6.4. Intrusion Detection

With the existence of commercial general-purpose components and current communication protocols, PLC-based control systems are threatened by cyberattacks. Fortunately, intrusion detection is a solution to the cybersecurity threats. Depending on data sources, intrusion detection can be classified into network-based and host-based detection. In this subsection, we discuss the ongoing methods targeting ICS in the respective categories. Main findings is featured in Table 3.

Table 3. Main findings in the intrusion detection.

Year	Data Source	Method	Security Focus	Ref.
2013	Traffic	DFA	Traffic abnormality	[87]
2014	Traffic	Control constraint modeling Autoregressive modeling	Variable abnormality	[88]
2016	Traffic	DFA	Sequence attacks	[89]
2017	Traffic	DFA	Sequence attacks	[90]
2017	Log	Decision trees SVM	Memory address values abnormality	[91]
2017	Log	OCSVM	PLC abnormal operation	[92]
2017	Field device status	Petri	PLC behavior abnormality	[93]
2018	Field device status	Petri	Sensor and actuator behavior abnormality	[94]
2019	HPC	SVM	HPC readings abnormality	[95]
2019	PLC parameters	FSM, (k,l)	PLC disruption	[96]

DFA: Deterministic Finite Automaton; SVM: Support Vector Machine; OCSVM: One-Class SVM; FSM: Finite State Machine; (k,l): a threshold signature scheme.

6.4.1. Network-Based Intrusion Detection

Network-based intrusion detection is necessary since cyberattack vectors are always hidden in the flow of network commands. There exists a kind of attack that contains multiple control commands, which are seemingly licit when viewed separately per packet, while they possibly breach the running states of the control systems. To detect the attacks, a critical-state analysis is presented to track a chain of packets changing the system states.

Furthermore, Deterministic Finite Automata (DFA) is a common method to provide a detailed traffic model for intrusion detection. For highly periodic Modbus/TCP traffic between HMIs and PLCs, Goldenberg and Wool developed DFA for each communication channel to detect anomalies sensitively [87]. Based on the previous work, Faisal et al. [89] proposed a complementary approach that combined the DFA with configuration-level specifications to monitor the communication. It effectively solved the problem that an appropriate amount of training data was required when retraining the model after the configuration changed. Interestingly, Markman et al. [90] found that the HMI-PLC channel was filled with bursts of packets relating to semantic meaning. Thus, they suggested a new burst-DFA model to detect anomalies in the traffic, which fitted the data much better compared with previous work.

Researchers studied the targeting control process in the PLC-based control system through the detection of semantic attacks. The attacks were divided into three sub-types: reconnaissance, direct control, and indirect control [88]. As a result, a semantic, network-based intrusion detection was presented by building a behavior model including constant, attribute, and continuous series. Deriving a set of expected values was used to model

constant and attribute data. Modeling the continuous data leveraged techniques of autoregression and control limits.

6.4.2. Host-Based Intrusion Detection

Researchers model state values for intrusion detection, including the values of relevant PLC memory addresses or state transitions of the control system. Supervised and semi-supervised machine learning methodologies were used separately to identify exceptions or abnormal behavior [91,92]. Leveraging the values of relevant memory addresses along with timestamps, the model could be used to distinguish abnormal PLC operations. Introducing the concept of controller behavior whitelisting, two Petri Net (PN)-based anomaly detection approaches were proposed from the perspective of experimental validation. Firstly, they manually constructed a white list, which was modeled on the field devices by the PN, and then converted it to the LD with a constraint condition of the PN, which allowed the PLC to detect abnormal behaviors [93]. However, there is a limitation to whitelisting in a manual way when the PLC-based control system is complex. To cope with the problem, they further presented an automatic generation method in which the representation of the whitelist used the SFC instead of the LD [94].

In addition, self-parameters can be applied to detect malicious attacks. Hardware performance counters are a series of registers to maintain low-level hardware events, e.g., the number of instructions retired, cache operations and exceptions, and the number of branches taken. Krishnamurthy et al. [95] modeled baseline behavior and then detected anomalies. It was suitable for the typical multi-threaded and interrupt-driven processes of PLCs. Combined with more parameters, a (k, l)-threshold signature scheme was developed by Chatterjee et al. [96] by using a finite state machine. What stood out was that it could detect both corrupted PLCs and compromised states within their proposed protocol, especially for legacy PLCs.

From the perspective of engineering practice, the following four aspects need to be considered in terms of safety-reliability metrics while performing intrusion detection: (1) It is better to use a local data archive to analyze the process data. By comparing aggregated values (per period, per process cycle) with the overall aggregated values from the archive, the significant difference in values is a potential feature for detection; (2) The allowable range for setting a timer or counter must be limited to meet operational requirements. If a remote tool such as an HMI writes values to a timer or counter, it is feasible to detect the presets and timeout values in the PLC before the writing action; (3) Similarly, a default value in an acceptable operating range is configured for each input variable that does not adversely affect the process and can be used as a flag for warnings. If a PLC variable receives an out-of-bounds value, the last valid value is entered for this variable, and an event is logged for further detection analysis; (4) Paired inputs or outputs are signals that cannot physically turn on at the same time; they exclude each other. Direct validation of the status of the paired inputs in the PLC is effective in detecting fault or malicious activity.

6.5. Honeypot-Based Detection

Different from the passive methods described in the previous subsections, honeypot-based detection is a technique that is employed to monitor network state, collect data, and analyze threats. In relation to the security of PLC-based control systems, there are various honeypot features such as obfuscation, high-fidelity emulation, secure malware storage, and traffic redirection. Thanks to the results from honeypots, malicious intrusions and potential exploitations can be revealed before fatal attacks are launched. Hence, some researchers' interest in security gradually shifts to honeypots for the PLC and its networks.

According to the interaction ability, the existing honeypots for PLCs could be mainly classified into two categories: low-interactive honeypots such as Conpot [97], as well as high-interactive honeypots such as CryPLH [98,99], XPOT [100], and S7COMMTrace [101]. When honeypots are applied for ICS, researchers take into account the following key characteristics: performance, authenticity, scalability, cost, and risk. Although the Conpot

supports seven kinds of protocols and limited function codes, it is easy for attackers to discover its fingerprint. To enhance the authenticity, CryPLH improved the interaction and added more original PLC implements. It was proven that the CryPLH was effectively deployed in real control networks to collect data. Compared with the CryPLH, S7COMMTrace had more sub-function codes in the protocol and higher fidelity simulation of PLCs, which reduced the risk of being discovered by cyberspace search engines like Shodan. Moreover, to achieve the further improvement of interactive capabilities, the PLC honeypot should support program compilation and interpretation. XPOT made it possible for the honeypot to be programmed with standard Integrated Development Environments (IDEs).

In short, we have discussed mainstream security detection schemes in the ongoing literature. Objectively speaking, there are also other security schemes to protect the ICS. For example, it is universally recognized that common PLC-based control systems are vulnerable to some types of access control attacks [102]. Scholars might suggest the introduction of an entropy algorithm in the data transmission message part [103] or a challenge-response authentication mechanism [104]. However, the defense resources are ultimately limited, especially for the current control systems. The impacts of traffic entropy on the current ICS were also discussed, including high cost, undesirable performance, and so on [105]. Besides, prevention capabilities against new threats or new attacks are weak because there are not enough security assessments. To some extent, some detection schemes have a more critical role than others [106]. That is the reason why we emphasize security detection schemes in this subsection. We believe that the advanced defense techniques could be applied to future PLC-based control systems, which will be discussed in Section 8.

7. Digital Forensic

Defense mechanisms are constructed to not only protect the PLC-based control systems from malicious intrusions or even attacks, but also be capable of taking precautions and tracing back to the root cause. Digital forensics plays a vital role in the latter aspect, which is one of the purposes of forensic science: to restore and investigate data remaining on digital devices. Although numerous frameworks, methodologies, and tools are developed for the forensics of IT systems, only a fraction of them could be directly applied for PLC-based control environments due to the main differences in the special requirements of the ICS. Researchers and practitioners face an array of challenges in the digital forensics of PLC-based control systems. Hence, we present the challenges and recent approaches in this section.

7.1. Challenges

As the requirements discussed in Section 2 demonstrate, there is a close connection between the ICS and the real physical world. Considering the continuity of the control process, the sensitivity of the time delay, the frequency of interaction, and the diversity of equipment, these external factors make digital forensics challenging. Besides, regarding the forensic-based techniques themselves, it is also supposed to take several key steps into consideration for the whole digital investigation, including data acquisition, evidence retention, analysis methods, simulation scenarios, and so on. It raises further challenges for digital forensics. Therefore, we summarize and highlight the challenges [5].

7.1.1. Challenges within Devices

1. Constrained resources: PLC-based control systems have constrained data-handling capacity with limited CPU, memory, I/O, and so on, especially for the legacy ones. Forensic investigators might encounter difficulties in data acquisition and analysis;
2. Local access: Due to the fact that field devices are distributed in remote regions, it is difficult for forensic tools that need local access to the compromised ones;
3. Proprietary systems: Vendor-specific devices apply proprietary protocols, operating systems, or even hardware that hinders generalized forensic tools in industrial control

environments. Some vendors usually provide a few appropriate interfaces for digital forensic functionality;
4. Insufficient logging: Because controlling and monitoring processes are their primary use, the relevant logging cannot support a holistic security investigation. Moreover, logging storage also adds burden to the poor memory of certain devices;
5. Mass process data: A large amount of lower control process data is generated from multiple sensors or actuators, which complicates the filtering and analyzing of valid data.

7.1.2. Challenges within Research

1. Simulation scenarios: With the help of simulators, high-fidelity industrial scenarios are still hard to repeat for digital forensic research experiments. Unfortunately, simulation without thorough consideration sometimes misleads the investigators, resulting in false judgment;
2. Small-scale testbeds: It is a wise choice for researchers to construct testbeds that consist of real physical equipment; however, it is also poorly scalable because of the high expenses;
3. Research for specific control processes: A distinct difference from digital forensics in IT systems is the specific control process. Nevertheless, neglected research in this area leads to a failure to trace back incidents that are brought on by a type of specialized attack such as semantic attacks.

7.1.3. Challenges within Human Factors

1. Lack of background knowledge: In the short term, it is rare for investigators to have background knowledge including complex control processes, details of compromised devices, the impact of forensic tools on performance, etc;
2. Industry collaboration: In consideration of data leakage, the majority of industrial enterprises refuse to cooperate with the research community. It may become an obstacle for the development of digital forensic tools and methodologies in practice.

7.2. Focus

According to the challenges, our survey focuses on the research of digital forensics for typical PLC-based control systems. Relevant literature inevitably covers methodologies, frameworks, and tools of data acquisition to some extent. Due to the 24/7 availability of the control systems, researchers have a deep insight into live data acquisition from running ones, such as volatile memory data and data on hard disk. Live forensics is gradually becoming a viable solution for real-time digital investigation. Besides, improving logging capabilities is also an effective approach to supporting digital evidence analysis. Sharing a set of similar goals with digital forensics, the incident response research focuses additionally on recovery from an ICS incident and restoring normal operations. From a practical point of view, both methodologies and developing tools combined with case studies have more application value. Particularly, it deserves to be studied how these achievements in digital forensics can be adapted to legacy PLC-based control systems. More detailed information is shown in Table A3.

7.3. Recent Approaches

Beyond summarizing the focus, we now present recent approaches in the domain of digital forensics for PLC-based control systems. These state-of-the-art approaches are applied at the device and network levels, respectively. At the network level, parsing proprietary industrial protocols and extracting data are common tendencies to analyze the network communication among devices [43,107–109]. At the device level, potential research is targeting the direct monitoring of the values of multiple PLC memory addresses [110,111]. Nonetheless, most of them are addressed to specific-vendor PLCs, such as those from Siemens, General Electric (GE), and Rockwell Allen-Bradley. Researchers strive to enhance

the generality of the proposed approaches for diverse devices and protocols. Taking the work of Choi et al. [112] as an example, with the help of PLC providing a web interface, a vendor-independent monitoring system was developed to collect security logs that could be used for digital forensics.

8. Future Work

Note that we listed considerable detection schemes and digital forensics methodologies in order to improve the cyber security of the current PLC-based control system. However, they were not enough to meet increasing demands to construct a robust, reliable, and advanced PLC-based control system. In this section, we highlight six main directions on security oriented toward future PLC-based control systems, which may differ from the current ones in several dimensions, including constituent parts, computing resources, core technologies, and application strategies. Figure 6 illustrates our outlined relationship among each part of the future work. One of the most straightforward ideas is to upgrade or update the control systems, such as through the utilization of secure embedded systems and communication protocols. Due to the need for testing and analysis, the control systems require validation capabilities with a high degree of fidelity, which can be realized by virtualization and open-source industrial control units. Furthermore, the integration of cloud-based, fog-based, and dynamic network techniques into the systems gives more opportunities for detection and defense against a series of ongoing security challenges.

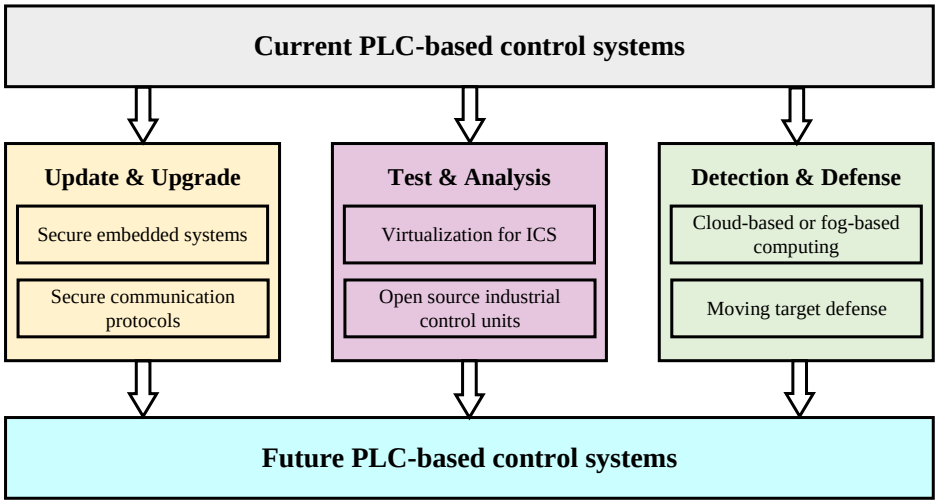

Figure 6. Relationship among each part of future work.

8.1. Secure Embedded Systems

The embedded system-based devices, which are autonomous, intelligent, and highly connected, have a principal role in the PLC-based control system. Nevertheless, the more powerful capabilities these systems obtain, the more sophisticated cyberattacks they will suffer from. Just like the safety PLC, a pre-defined reaction on failure was designed in the PLC to fulfill reliable and safe requirements for industrial processes. In the future, more built-in security features should be supported for embedded systems, so-called "secure embedded systems". The vulnerabilities of current PLCs themselves mostly lie in their firmware or control logic programs, as mentioned in Section 4. For that reason, new advanced secure PLCs need to consider security properties such as secure boot, secure program update, and embedded management. The novel proof of concept of controllers with embedded hypervisors, CPUs with security processors, a secure firmware update framework, and even a software- or hardware-based secure boot mechanism will be realized in the next-generation PLC-based control systems where the Design Lifecycle of Secure

Embedded Devices System approach is applied. However, it simultaneously means that it will cost more computational power. An open issue is how to ensure security while maintaining normal functionality.

8.2. Secure Communication Protocols

Obviously, the security of existing PLC-based control systems is subject to the defects of legacy communication protocols, including plaintext transmission and the lack of user authentication mechanisms as well as integrity checking. As mentioned in Section 4, the systems are prone to being violated by attacks such as DoS, injection, replay, and MITM. As a result, extensive industrial communication protocol update versions emerged in diverse research fields, i.e., S7COMM-Plus and Modbus/TCP Security. To a large extent, these variant versions are not suitable for current resource-constrained control systems to achieve satisfactory real-time or stable performance, even if targeted adjustments are made in relevant software or hardware. What is worse, some secure-version protocols can still be exploited by sophisticated attackers. For instance, S7COMMPlus uses private algorithms to encrypt connection packets and function packets, which can effectively protect the communication between PLCs and the TIA portal from replay attack. Whereas, by means of reverse engineering the cryptographic protocol, adversaries also have a chance to launch a protocol-oriented attack. Therefore, a hot-spot research topic may be the favorable improvements of corresponding industrial communication protocols or innovatively designed secure ones.

8.3. Virtualization for ICS

Considering the cost and industry practices, it is impossible to analyze experiments or validate security solutions in real plants. Hence, there is an urgent demand for test environments, namely testbeds. The testbeds probably can be divided into four categories, including implementation-based ones, simulations with actual devices, single-simulations, and federated simulations. Due to the rise of virtualization techniques, virtual federated simulations for ICS have cost-efficient and scalable advantages over other kinds. Main research interest focuses on the virtualization of industry devices, such as virtual PLCs, as well as the construction honeypot with virtual hosts in the industrial control network. However, issues related to the fidelity of virtual devices should be taken into further consideration. For example, once attackers find out that the fingerprints of virtual devices in the honeypot differ from the real ones, they may not conduct the next violation steps in the virtual environment. In other words, the honeypot loses its value in some deceptive ways because of the failure in virtual device fingerprint emulation. In a broad sense, the future virtualization of ICS is not only for constructing a preferable performance testbed but also for pursuing an outstanding and manageable substitution for the current industrial devices.

8.4. Open-Source Industrial Control Units

A challenge of emulating PLCs in virtual environments could be missing information about their internal behaviors, especially by simply porting the PLC source codes to runtime on general-purpose operating systems. To obtain the actual response value while conducting a simulated attack experiment and exploring weaknesses in the system, researchers have to use PLCs as "hardware-in-the-loop". Unfortunately, the proprietary software and hardware that vendors offer hinder further security research on the surrounding parts of the embedded systems, such as the operational logic and internal mechanisms. To get rid of the above dilemma, the OpenPLC project was proposed to build a functional, standardized open-source paradigm that provides the entire source code, the IDE, and the available hardware configurations on the Raspberry Pi, Arduino, and ESP8266. The modular framework in the OpenPLC project allows the researchers to construct self-defined testbeds with a combination of virtualization according to their needs, like designing an encryption layer built-in to PLCs to directly secure communication channels. In the future,

apart from open-source PLCs, increasing approaches to open-source industrial control units ought to be developed for the research community to accomplish more targeted cybersecurity analysis for each key component of PLC-based control systems.

8.5. Cloud-Based or Fog-Based Computing

Within the structure of the Industrial Internet of Things, cloud-based or fog-based computing has enabled the migration of the classical PLC-based control systems to the cloud. It demonstrates an adaptable and agile solution to treating PLCs as a service. Meanwhile, attack surfaces are expanding owing to new vulnerabilities in complex and large-scale frameworks and initial access from multiple sources. Future concerns about protecting such control systems will be implemented by security platforms on the cloud. Take the following innovatively creative platforms as examples: A prototype, called PLC-Cloud, performed heavy security analysis leveraging cloud computing resources to offer a last-step verification of commands before reacting to actual controlled devices. For Service-Oriented Architecture systems, a security cloud platform was built to assure data integrity and avoid the risk of failures and attacks by providing a "toolbox" with the functions of service planning, end-to-end security, and monitoring and policing. Subsequently, the toolbox was applied to strengthen the security and privacy at the fog layer to counter cyberattacks like the compromised fog node attack. In brief, these future security platforms have common features such as low resource consumption in the local industrial control context, which reduces the extra performance effect such as latency delay. To that end, more advanced mechanisms could be deployed on the cloud to enhance the global security of the next-generation systems.

8.6. Moving Target Defense

In terms of the requirements for ICS given in Section 2, it is impossible to update or patch the systems frequently, especially the highly-continuous processing ones. MTD is another potential active defense technique to allow a portion of unpatched vulnerabilities to exist in the underlying systems and looks forward to providing extra mechanisms that add difficulties for adversaries while launching attacks. So far, there have been two representative MTD researches for the control systems, consisting of dynamic IP and random configure parameters. By using dynamic IP techniques, peer hosts could not be easily recognized at the reconnaissance stage, and communication between them was also difficult. The random configuration parameters make the systems time-varying and stochastic, so that the specific knowledge that attackers collect beforehand about the control process is limited. Note that MTD is not an independent and complete scheme but a redundant and coordinated security one. Therefore, MTD may be an effective way to secure the future of PLC-based control systems by improving their resiliency.

9. Conclusions

The comprehensive literature review focused on the security of PLC-based control systems; it surveyed vulnerabilities, attacks, security detection schemes, digital forensic research, and future works from two aspects: the core component level and the system level. We did not only take consideration of current control system security but also offer recommendations for future control systems. Compared with the existing surveys, our work proposed specific classifications for vulnerabilities, attacks, and security detection schemes. Our analysis of PLCs' vulnerabilities contained program, memory, and firmware. In addition, application software, communication protocols, and connected devices related to relevant control systems were researched too. Subsequently, the existing attacks are classified into three categories, including attacks against availability, integrity, and confidentiality. Then, we presented security detection schemes with detection functions, which are classified as program detection, firmware detection, device-fingerprinting-based detection, intrusion detection, and honeypot-based detection. Furthermore, we discussed methodology, challenges, and achievements in forensic research. With an eye toward

further research, suggestions were outlined to strengthen security in future PLC-based control systems. In the light of mitigating cyber threats to ICS targets, sound mechanisms are needed for overall secure control systems due to the fact that single security appliances and technologies are somewhat limited.

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

Funding: Our work is supported by the National Key R&D Program of China (2021YFB2012400).

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

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

Appendix A

Table A1. Comparison with existing surveys.

Year	Ref.	Level	Sou.		Vul.		Att.		Sec.	
			DB	Type	Invo.	Categ.	Invo.	Categ.	Invo.	Categ.
2012	Milinkovic and Lazic [8]	C	IEEE	C	✓	I			✓	I
2016	McLaughlin et al. [7]	S	IEEE	M	✓	C	✓	C	✓	C
2016	Khorrami et al. [9]	S	IEEE	C	✓	I	✓	I	✓	I
2016	Amrein et al. [10]	S	IEEE	M					✓	I
2017	Rubio et al. [11]	S	SCITE	C					✓	C
2017	Nazir et al. [12]	S	Elsevier	M	✓	C	✓	C	✓	C
2018	Davidson [13]	C	ACI	C	✓	I	✓	I	✓	I
2018	Hu et al. [14]	S	SAGE	M	✓	I			✓	C
2019	Rodofile et al. [15]	S	Elsevier	M			✓	C		
2019	Volkova et al. [16]	S	IEEE	M			✓	C	✓	I
2020	Pan et al. [17]	C	TSP	M	✓	I	✓	I	✓	I
2020	Bhamare et al. [18]	S	Elsevier	M					✓	I
2021	Sun et al. [19]	C	IEEE	C			✓	I	✓	I
2021	Nguyen et al. [20]	S	Springer	C	✓	I	✓	I	✓	I
2021	Hajda et al. [21]	S	MDPI	M					✓	C
2022	Alanazi et al. [22]	S	Elsevier	M	✓	C	✓	C	✓	C
2022	Koay et al. [23]	S	Springer	M	✓	I	✓	I	✓	I
2022	Kayan et al. [24]	S	ACM	M	✓	C	✓	C	✓	C

(1) Ref.: Reference; Sou.: Source of literature; Vul.: Vulnerablities; Att.: Attacks; Sec.: Security Schemes; Invo.: Involving; Categ.: Categories; (2) In Level column, S: system-level, C: component-level; Sou. DB: search database; In Sou. Type column, C: conference, M: magazine; (3) In Categ. column of Vul., Att. and Sec., I: an incompletely analysis, C: a comprehensive analysis, <empty>: not mention.

Table A2. List of acronyms.

Acronym	Description	Acronym	Description
ARP	Address resolution protocol	ARCADE	Aachen rigorous code analysis and debugging environment
CFG	Control flow graph	CFI	Control flow integrity
CPU	Central processing unit	DEO	Denial of engineering operation
DoS	Denial of service	DFA	Deterministic finite automata
EM	Electromagnetic	ENIP	EtherNet industrial protocol
FBD	Function block diagram	GRE	Generic routing encapsulation

Table A2. *Cont.*

Acronym	Description	Acronym	Description
HMI	Human–machine interface	ICS	Industrial control system
IDE	Integrated development environment	IT	Information technology
IL	Instruction list	IP	Internet protocol
LD	Ladder diagram	MAC	Media access control
MITM	Man-in-the-middle	MTD	Moving target defense
OS	Operating system	PCCC	Programmable controller communication command
PLC	Programmable logic controller	PN	Petri net
SCADA	Supervisory control and data acquisition systems	RF	Radio frequency
SFC	Sequential function char	ST	Structured text
SYN	Synchronize sequence number	TCP	Transmission control protocol
TIA	Total integrated automation	UDP	User datagram protocol

Table A3. Focus on digital forensics for the PLC-based control systems.

Authors	Level	PLC Info.	Data	Method
Kleinmann et al. [107]	Network	Siemens PLC	Network traffic	Model with DFA
Wu et al. [110]	Device	Siemens S7-1200	Values of relevant memory addresses	1. Acquire the program code using PLC Logger 2. Use an existing tool CFTT to test PLC Logger's suitability
Yau et al. [111]	Device	Siemens S7-1200	Values of relevant memory addresses	With a set of Detection Rules to detect and record undesired incidents
Chan et al. [113]	Device	Siemens PLC	TIA portal project file	Analyze with the TIA portal
Denton et al. [109]	Network	GE Fanuc Series 90-30	Values of relevant memory addresses	1. Communicate with the PLC directly 2. Read memory and identifying attacks
Senthivel et al. [43]	Network	Allen-Bradley Micrologix 1400	Network traffic	1. Prase the PCCC protocol 2. Analyze the extracted file
Yau et al. [91]	Device	Siemens S7-1200	Values of relevant memory addresses	Classify with an One-Class Support Vector Machine algorithm
Yau et al. [92]	Device	Siemens S7-1200	Values of relevant memory addresses	Classify with a decision tree algorithm and a Support Vector Machine algorithm
Chan et al. [114]	Device	Siemens S7-1200	1. critical-value 2. data-block-address 3.timestamp	Incorporate a security block in a PLC
Yau et al. [108]	Network	Siemens S7-1200	Network traffic	Extracts data from Siemens S7 communication protocol traffic
Choi et al. [112]	Network	PLC providing web interface	System information via a web interface	Collect security logs for PLCs supporting the web interface

References

1. Falliere, N.; Murchu, L.O.; Chien, E. W32. stuxnet dossier. *White Pap. Symantec Corp. Secur. Response* **2011**, *5*, 29.
2. Liang, G.; Weller, S.R.; Zhao, J.; Luo, F.; Dong, Z.Y. The 2015 ukraine blackout: Implications for false data injection attacks. *IEEE Trans. Power Syst.* **2016**, *32*, 3317–3318. [CrossRef]
3. Akbanov, M.; Vassilakis, V.G.; Logothetis, M.D. Ransomware detection and mitigation using software-defined networking: The case of WannaCry. *Comput. Electr. Eng.* **2019**, *76*, 111–121. [CrossRef]
4. Di Pinto, A.; Dragoni, Y.; Carcano, A. TRITON: The first ICS cyber attack on safety instrument systems. In Proceedings of the Black Hat USA 2018, Las Vegas, NV, USA, 4–9 August 2018; pp. 1–26.
5. Ahmed, I.; Obermeier, S.; Sudhakaran, S.; Roussev, V. Programmable logic controller forensics. *IEEE Secur. Priv.* **2017**, *15*, 18–24. [CrossRef]
6. Fronchetti, F.; Ritschel, N.; Holmes, R.; Li, L.; Soto, M.; Jetley, R.; Wiese, I.; Shepherd, D. Language impact on productivity for industrial end users: A case study from Programmable Logic Controllers. *J. Comput. Lang.* **2022**, *69*, 101087. [CrossRef]
7. McLaughlin, S.; Konstantinou, C.; Wang, X.; Davi, L.; Sadeghi, A.R.; Maniatakos, M.; Karri, R. The cybersecurity landscape in industrial control systems. *Proc. IEEE* **2016**, *104*, 1039–1057. [CrossRef]
8. Milinković, S.A.; Lazić, L.R. Industrial PLC security issues. In Proceedings of the 2012 20th Telecommunications Forum (TELFOR), Belgrade, Serbia, 20–22 November 2012; pp. 1536–1539.
9. Khorrami, F.; Krishnamurthy, P.; Karri, R. Cybersecurity for control systems: A process-aware perspective. *IEEE Des. Test* **2016**, *33*, 75–83. [CrossRef]
10. Amrein, A.; Angeletti, V.; Beitler, A.; Német, M.; Reiser, M.; Riccetti, S.; Stoecklin, M.P.; Wespi, A. Security intelligence for industrial control systems. *IBM J. Res. Dev.* **2016**, *60*, 11–13. [CrossRef]
11. Rubio, J.E.; Alcaraz, C.; Roman, R.; Lopez, J. Analysis of Intrusion Detection Systems in Industrial Ecosystems. In Proceedings of the SECRYPT, Madrid, Spain, 26–28 July 2017; pp. 116–128.
12. Nazir, S.; Patel, S.; Patel, D. Assessing and augmenting SCADA cyber security: A survey of techniques. *Comput. Secur.* **2017**, *70*, 436–454. [CrossRef]
13. Davidson, C.C.; Andel, T.; Yampolskiy, M.; McDonald, J.T.; Glisson, B.; Thomas, T. On SCADA PLC and Fieldbus Cyber-Security. In Proceedings of the 13th International Conference on Cyber Warfare and Security, Washington, DC, USA, 8–9 March 2018; pp. 140–149.
14. Hu, Y.; Yang, A.; Li, H.; Sun, Y.; Sun, L. A survey of intrusion detection on industrial control systems. *Int. J. Distrib. Sens. Netw.* **2018**, *14*, 1550147718794615. [CrossRef]
15. Rodofile, N.R.; Radke, K.; Foo, E. Extending the cyber-attack landscape for SCADA-based critical infrastructure. *Int. J. Crit. Infrastruct. Prot.* **2019**, *25*, 14–35. [CrossRef]
16. Volkova, A.; Niedermeier, M.; Basmadjian, R.; de Meer, H. Security challenges in control network protocols: A survey. *IEEE Commun. Surv. Tutor.* **2018**, *21*, 619–639. [CrossRef]
17. Pan, X.; Wang, Z.; Sun, Y. Review of PLC security issues in industrial control system. *J. Cybersecur.* **2020**, *2*, 69. [CrossRef]
18. Bhamare, D.; Zolanvari, M.; Erbad, A.; Jain, R.; Khan, K.; Meskin, N. Cybersecurity for industrial control systems: A survey. *Comput. Secur.* **2020**, *89*, 101677. [CrossRef]
19. Sun, R.; Mera, A.; Lu, L.; Choffnes, D. SoK: Attacks on industrial control logic and formal verification-based defenses. In Proceedings of the 2021 IEEE European Symposium on Security and Privacy (EuroS&P), Vienna, Austria, 6–10 September 2021; pp. 385–402.
20. Nguyen, H.P.D.; Ruiz, L.; Rajnai, Z. Industrial Control System (ICS): The General Overview of the Security Issues and Countermeasures. In *Informatics and Cybernetics in Intelligent Systems: Proceedings of 10th Computer Science On-Line Conference 2021*; Springer: Cham, Switzerland, 2021; Volume 228, pp. 412–419.
21. Hajda, J.; Jakuszewski, R.; Ogonowski, S. Security Challenges in Industry 4.0 PLC Systems. *Appl. Sci.* **2021**, *11*, 9785. [CrossRef]
22. Alanazi, M.; Mahmood, A.; Chowdhury, M.J.M. SCADA Vulnerabilities and Attacks: A Review of the State-of-the-Art and Open Issues. *Comput. Secur.* **2023**, *125*, 103028. [CrossRef]
23. Koay, A.M.; Ko, R.K.L.; Hettema, H.; Radke, K. Machine learning in industrial control system (ICS) security: Current landscape, opportunities and challenges. *J. Intell. Inf. Syst.* **2022**, 1–29. [CrossRef]
24. Kayan, H.; Nunes, M.; Rana, O.; Burnap, P.; Perera, C. Cybersecurity of industrial cyber-physical systems: A review. *ACM Comput. Surv. (CSUR)* **2022**, *54*, 1–35. [CrossRef]
25. Serhane, A.; Raad, M.; Raad, R.; Susilo, W. PLC code-level vulnerabilities. In Proceedings of the 2018 International Conference on Computer and Applications (ICCA), Beirut, Lebanon, 25–26 August 2018; pp. 348–352.
26. Serhane, A.; Raad, M.; Raad, R.; Susilo, W. Programmable logic controllers based systems (PLC-BS): Vulnerabilities and threats. *SN Appl. Sci.* **2019**, *1*, 924. [CrossRef]
27. Govil, N.; Agrawal, A.; Tippenhauer, N.O. On ladder logic bombs in industrial control systems. In Proceedings of the Computer Security, Oslo, Norway, 14–15 September 2017; pp. 110–126.
28. Bonney, G.; Höfken, H.; Paffen, B.; Schuba, M. ICS/SCADA security analysis of a beckhoff CX5020 PLC. In Proceedings of the 2015 International Conference on Information Systems Security and Privacy (ICISSP), Angers, France, 9–11 February 2015; pp. 1–6.

29. 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.
30. Leverett, É.; Wightman, R. Vulnerability inheritance programmable logic controllers. In Proceedings of the Second International Symposium on Research in Grey-Hat Hacking, Grenoble, France, 15 November 2013.
31. Hui, H.; McLaughlin, K. Investigating current plc security issues regarding siemens s7 communications and TIA portal. In Proceedings of the 5th International Symposium for ICS & SCADA Cyber Security Research 2018, Hamburg, Germany, 29–30 August 2018; pp. 67–73.
32. Basnight, Z.; Butts, J.; Lopez, J., Jr.; Dube, T. Firmware modification attacks on programmable logic controllers. *Int. J. Crit. Infrastruct. Prot.* **2013**, *6*, 76–84. [CrossRef]
33. Garcia, L.; Brasser, F.; Cintuglu, M.H.; Sadeghi, A.R.; Mohammed, O.A.; Zonouz, S.A. Hey, My Malware Knows Physics! Attacking PLCs with Physical Model Aware Rootkit. In Proceedings of the NDSS, San Diego, CA, USA, 26 February–1 March 2017.
34. Robles-Durazno, A.; Moradpoor, N.; McWhinnie, J.; Russell, G.; Maneru-Marin, I. Implementation and Detection of Novel Attacks to the PLC Memory of a Clean Water Supply System. In Proceedings of the International Conference on Technology Trends CITT 2018, Babahoyo, Ecuador, 29–31 August 2018; Springer: Cham, Switzerland, 2018; pp. 91–103.
35. 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. Infrastruct. Prot.* **2019**, *26*, 100300. [CrossRef]
36. Tacliad, F.; Nguyen, T.D.; Gondree, M. DoS Exploitation of Allen-Bradley's Legacy Protocol through Fuzz Testing. In Proceedings of the 3rd Annual Industrial Control System Security Workshop, San Juan, PR, USA, 5 December 2017; pp. 24–31.
37. 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.
38. Sayegh, N.; Chehab, A.; Elhajj, I.H.; Kayssi, A. Internal security attacks on SCADA systems. In Proceedings of the 2013 Third International Conference on Communications and Information Technology (ICCIT), Beirut, Lebanon, 19–21 June 2013; pp. 22–27.
39. Niedermaier, M.; Malchow, J.O.; Fischer, F.; Marzin, D.; Merli, D.; Roth, V.; Von Bodisco, A. You snooze, you lose: Measuring {PLC} cycle times under attacks. In Proceedings of the 12th {USENIX} Workshop on Offensive Technologies ({WOOT} 18), Baltimore, MD, USA, 13–14 August 2018.
40. Yang, H.; Cheng, L.; Chuah, M.C. Detecting payload attacks on programmable logic controllers (plcs). In Proceedings of the 2018 IEEE Conference on Communications and Network Security (CNS), Beijing, China, 30 May–1 June 2018; pp. 1–9.
41. McLaughlin, S.E. On Dynamic Malware Payloads Aimed at Programmable Logic Controllers. In Proceedings of the HotSec, San Francisco, CA, USA, 9 August 2011
42. McLaughlin, S.; McDaniel, P. SABOT: Specification-based payload generation for programmable logic controllers. In Proceedings of the 2012 ACM Conference on Computer and Communications Security, Raleigh, NC, USA, 16–18 October 2012; pp. 439–449.
43. Senthivel, S.; Ahmed, I.; Roussev, V. SCADA network forensics of the PCCC protocol. *Digit. Investig.* **2017**, *22*, S57–S65. [CrossRef]
44. Qasim, S.A.; Lopez, J.; Ahmed, I. Automated Reconstruction of Control Logic for Programmable Logic Controller Forensics. In Proceedings of the International Conference on Information Security, ISC 2019, New York, NY, USA, 16–18 September 2019; Springer: Cham, Switzerland, 2019; pp. 402–422.
45. Yoo, H.; Kalle, S.; Smith, J.; Ahmed, I. Overshadow PLC to detect remote control-logic injection attacks. In Proceedings of the International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, DIMVA 2019, Gothenburg, Sweden, 19–20 June 2019; Springer: Cham, Switzerland, 2019; pp. 109–132.
46. McLaughlin, S.; Zonouz, S. Controller-aware false data injection against programmable logic controllers. In Proceedings of the 2014 IEEE International Conference on Smart Grid Communications (SmartGridComm), Venice, Italy, 3–6 November 2014; pp. 848–853.
47. Xiao, M.; Wu, J.; Long, C.; Li, S. Construction of false sequence attack against PLC based power control system. In Proceedings of the 2016 35th Chinese Control Conference (CCC), Chengdu, China, 27–29 July 2016; pp. 10090–10095.
48. Fritz, R.; Schwarz, P.; Zhang, P. Modeling of Cyber Attacks and a Time Guard Detection for ICS based on Discrete Event Systems. In Proceedings of the 2019 18th European Control Conference (ECC), Naples, Italy, 25–28 June 2019; pp. 4368–4373.
49. Yoo, H.; Ahmed, I. Control logic injection attacks on industrial control systems. In Proceedings of the IFIP International Conference on ICT Systems Security and Privacy Protection, SEC 2019, Lisbon, Portugal, 25–27 June 2019; Springer: Cham, Switzerland, 2019; pp. 33–48.
50. Abbasi, A.; Hashemi, M.; Zambon, E.; Etalle, S. Stealth low-level manipulation of programmable logic controllers i/o by pin control exploitation. In Proceedings of the Critical Information Infrastructures Security: 11th International Conference, CRITIS 2016, Paris, France, 10–12 October 2016; pp. 1–12.
51. Abbasi, A.; Hashemi, M. Ghost in the plc designing an undetectable programmable logic controller rootkit via pin control attack. *Black Hat Eur.* **2016**, *2016*, 1–35.
52. Beresford, D. Exploiting siemens simatic s7 plcs. *Black Hat USA* **2011**, *16*, 723–733.

53. Eigner, O.; Kreimel, P.; Tavolato, P. Identifying S7comm Protocol Data Injection Attacks in Cyber-Physical Systems. In Proceedings of the 5th International Symposium for ICS & SCADA Cyber Security Research 2018, Hamburg, Germany, 29–30 August 2018; pp. 51–56.
54. Ghaleb, A.; Zhioua, S.; Almulhem, A. On PLC network security. *Int. J. Crit. Infrastruct. Prot.* **2018**, *22*, 62–69. [CrossRef]
55. Lim, B.; Chen, D.; An, Y.; Kalbarczyk, Z.; Iyer, R. Attack induced common-mode failures on plc-based safety system in a nuclear power plant: Practical experience report. In Proceedings of the 2017 IEEE 22nd Pacific Rim International Symposium on Dependable Computing (PRDC), Christchurch, New Zealand, 22–25 January 2017; pp. 205–210.
56. Grandgenett, R.; Mahoney, W.; Gandhi, R. Authentication bypass and remote escalated I/O command attacks. In Proceedings of the 10th Annual Cyber and Information Security Research Conference, Oak Ridge, TN, USA, 7–9 April 2015; pp. 1–7.
57. Lei, C.; Donghong, L.; Liang, M. The spear to break the security wall of S7CommPlus. *Blackhat USA* **2017**, *17*, 1–12.
58. Klick, J.; Lau, S.; Marzin, D.; Malchow, J.O.; Roth, V. Internet-facing PLCs as a network backdoor. In Proceedings of the 2015 IEEE Conference on Communications and Network Security (CNS), Florence, Italy, 28–30 September 2015; pp. 524–532.
59. Spenneberg, R.; Brüggemann, M.; Schwartke, H. Plc-blaster: A worm living solely in the plc. *Black Hat Asia* **2016**, *16*, 1–16.
60. Yao, Y.; Sheng, C.; Fu, Q.; Liu, H.; Wang, D. A propagation model with defensive measures for PLC-PC worms in industrial networks. *Appl. Math. Model.* **2019**, *69*, 696–713. [CrossRef]
61. Korkmaz, E.; Davis, M.; Dolgikh, A.; Skormin, V. Detection and mitigation of time delay injection attacks on industrial control systems with PLCs. In Proceedings of the Computer Network Security: 7th International Conference on Mathematical Methods, Models, and Architectures for Computer Network Security, MMM-ACNS 2017, Warsaw, Poland, 28–30 August 2017; Proceedings 7; Springer: Cham, Switzerland, 2017; pp. 62–74.
62. Falco, G.; Caldera, C.; Shrobe, H. IIoT cybersecurity risk modeling for SCADA systems. *IEEE Internet Things J.* **2018**, *5*, 4486–4495. [CrossRef]
63. Ayub, A.; Yoo, H.; Ahmed, I. Empirical study of PLC authentication protocols in industrial control systems. In Proceedings of the 2021 IEEE Security and Privacy Workshops (SPW), San Francisco, CA, USA, 27 May 2021; pp. 383–397.
64. Yang, K.; Wang, H.; Sun, L. An effective intrusion-resilient mechanism for programmable logic controllers against data tampering attacks. *Comput. Ind.* **2022**, *138*, 103613. [CrossRef]
65. Xiao, L.; Li, M.; Gu, M.; Sun, J. A hierarchy framework on compositional verification for PLC software. In Proceedings of the 2014 IEEE 5th International Conference on Software Engineering and Service Science, Beijing, China, 27–29 June 2014; pp. 204–207.
66. Stattelmann, S.; Biallas, S.; Schlich, B.; Kowalewski, S. Applying static code analysis on industrial controller code. In Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA), Barcelona, Spain, 16–19 September 2014; pp. 1–4.
67. Zhang, M.; Chen, C.Y.; Kao, B.C.; Qamsane, Y.; Shao, Y.; Lin, Y.; Shi, E.; Mohan, S.; Barton, K.; Moyne, J. Towards Automated Safety Vetting of PLC Code in Real-World Plants. In Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 19–23 May 2019; pp. 522–538.
68. Ovatman, T.; Aral, A.; Polat, D.; Ünver, A.O. An overview of model checking practices on verification of PLC software. *Softw. Syst. Model.* **2016**, *15*, 937–960. [CrossRef]
69. Xiao, L.; Wang, R.; Gu, M.; Sun, J. Semantic characterization of programmable logic controller programs. *Math. Comput. Model.* **2012**, *55*, 1819–1824. [CrossRef]
70. McLaughlin, S.E.; Zonouz, S.A.; Pohly, D.J.; McDaniel, P.D. A Trusted Safety Verifier for Process Controller Code. In Proceedings of the NDSS, San Diego, CA, USA, 23–26 February 2014; Volume 14.
71. Zonouz, S.; Rrushi, J.; McLaughlin, S. Detecting industrial control malware using automated PLC code analytics. *IEEE Secur. Priv.* **2014**, *12*, 40–47. [CrossRef]
72. Chang, T.; Wei, Q.; Liu, W.; Geng, Y. Detecting PLC Program Malicious Behaviors Based on State Verification. In Proceedings of the International Conference on Cloud Computing and Security, ICCCS 2018, Haikou, China, 8–10 June 2018; Springer: Cham, Switzerland, 2018; pp. 241–255.
73. Kottler, S.; Khayamy, M.; Hasan, S.R.; Elkeelany, O. Formal verification of ladder logic programs using NuSMV. In Proceedings of the SoutheastCon 2017, Concord, NC, USA, 30 March–2 April 2017; pp. 1–5.
74. Hailesellasie, M.; Hasan, S.R. Intrusion Detection in PLC-Based Industrial Control Systems Using Formal Verification Approach in Conjunction with Graphs. *J. Hardw. Syst. Secur.* **2018**, *2*, 1–14. [CrossRef]
75. Lv, X.; Xie, Y.; Zhu, X.; Ren, L. A technique for bytecode decompilation of PLC program. In Proceedings of the 2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chongqing, China, 25–26 March 2017; pp. 252–257.
76. Keliris, A.; Maniatakos, M. Icsref: A framework for automated reverse engineering of industrial control systems binaries. *arXiv* **2018**, arXiv:1812.03478.
77. Chang, T.; Wei, Q.; Geng, Y.; Zhang, H. Constructing PLC binary program model for detection purposes. *J. Phys. Conf. Ser.* **2018**, *1087*, 22022. [CrossRef]
78. Abbasi, A.; Holz, T.; Zambon, E.; Etalle, S. ECFI: Asynchronous control flow integrity for programmable logic controllers. In Proceedings of the 33rd Annual Computer Security Applications Conference, Orlando, FL, USA, 4–8 December 2017; pp. 437–448.
79. McMinn, L.; Butts, J. A firmware verification tool for programmable logic controllers. In Proceedings of the International Conference on Critical Infrastructure Protection, ICCIP 2012, Washington, DC, USA, 19–21 March 2012; Springer: Berlin/Heidelberg, Germany, 2012; pp. 59–69.

80. Stone, S.; Temple, M. Radio-frequency-based anomaly detection for programmable logic controllers in the critical infrastructure. *Int. J. Crit. Infrastruct. Prot.* **2012**, *5*, 66–73. [CrossRef]
81. Stone, S.J.; Temple, M.A.; Baldwin, R.O. Detecting anomalous programmable logic controller behavior using RF-based Hilbert transform features and a correlation-based verification process. *Int. J. Crit. Infrastruct. Prot.* **2015**, *9*, 41–51. [CrossRef]
82. Gonzalez, C.A.; Hinton, A. Detecting malicious software execution in programmable logic controllers using power fingerprinting. In Proceedings of the International Conference on Critical Infrastructure Protection, ICCIP 2014, Arlington, VA, USA, 17–19 March 2014; Springer: Berlin/ Heidelberg, Germany, 2014; pp. 15–27.
83. Xiao, Y.j.; Xu, W.y.; Jia, Z.h.; Ma, Z.r.; Qi, D.l. NIPAD: A non-invasive power-based anomaly detection scheme for programmable logic controllers. *Front. Inf. Technol. Electron. Eng.* **2017**, *18*, 519–534. [CrossRef]
84. Dunlap, S.; Butts, J.; Lopez, J.; Rice, M.; Mullins, B. Using timing-based side channels for anomaly detection in industrial control systems. *Int. J. Crit. Infrastruct. Prot.* **2016**, *15*, 12–26. [CrossRef]
85. Boggs, N.; Chau, J.C.; Cui, A. Utilizing electromagnetic emanations for out-of-band detection of unknown attack code in a programmable logic controller. In Proceedings of the Cyber Sensing 2018. International Society for Optics and Photonics, Orlando, FL, USA, 17–18 April 2018; Volume 10630, p. 106300D.
86. Van Aubel, P.; Papagiannopoulos, K.; Chmielewski, Ł.; Doerr, C. Side-channel based intrusion detection for industrial control systems. In Proceedings of the International Conference on Critical Information Infrastructures Security, CRITIS 2017, Lucca, Italy, 8–13 October 2017; Springer: Cham, Switzerland, 2017; pp. 207–224.
87. Goldenberg, N.; Wool, A. Accurate modeling of Modbus/TCP for intrusion detection in SCADA systems. *Int. J. Crit. Infrastruct. Prot.* **2013**, *6*, 63–75. [CrossRef]
88. Hadžiosmanović, D.; Sommer, R.; Zambon, E.; Hartel, P.H. Through the eye of the PLC: Semantic security monitoring for industrial processes. In Proceedings of the 30th Annual Computer Security Applications Conference, New Orleans, LA, USA, 8–12 December 2014; pp. 126–135.
89. Faisal, M.; Cardenas, A.A.; Wool, A. Modeling Modbus TCP for intrusion detection. In Proceedings of the 2016 IEEE Conference on Communications and Network Security (CNS), Philadelphia, PA, USA, 17–19 October 2016; pp. 386–390.
90. Markman, C.; Wool, A.; Cardenas, A.A. A new burst-DFA model for SCADA anomaly detection. In Proceedings of the 2017 Workshop on Cyber-Physical Systems Security and PrivaCy, Dallas, TX, USA, 3 November 2017; pp. 1–12.
91. Yau, K.; Chow, K.P. Detecting anomalous programmable logic controller events using machine learning. In Proceedings of the IFIP International Conference on Digital Forensics, DigitalForensics 2017, Orlando, FL, USA, 30 January–1 February 2017; Springer: Cham, Switzerland, 2017; pp. 81–94.
92. Yau, K.; Chow, K.P.; Yiu, S.M.; Chan, C.F. Detecting anomalous behavior of PLC using semi-supervised machine learning. In Proceedings of the 2017 IEEE Conference on Communications and Network Security (CNS), Las Vegas, NV, USA, 9–11 October 2017; pp. 580–585.
93. Mochizuki, A.; Sawada, K.; Shin, S.; Hosokawa, S. On experimental verification of model based white list for PLC anomaly detection. In Proceedings of the 2017 11th Asian Control Conference (ASCC), Gold Coast, QLD, Australia, 17–20 December 2017; pp. 1766–1771.
94. Fujita, S.; Rata, K.; Mochizuki, A.; Sawada, K.; Shin, S.; Hosokawa, S. On Experimental validation of Whitelist Auto-Generation Method for Secured Programmable Logic Controllers. In Proceedings of the IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society, Washington, DC, USA, 21–23 October 2018; pp. 2385–2390.
95. Krishnamurthy, P.; Karri, R.; Khorrami, F. Anomaly detection in real-time multi-threaded processes using hardware performance counters. *IEEE Trans. Inf. Forensics Secur.* **2019**, *15*, 666–680. [CrossRef]
96. Chatterjee, U.; Santikellur, P.; Sadhukhan, R.; Govindan, V.; Mukhopadhyay, D.; Chakraborty, R.S. United We Stand: A Threshold Signature Scheme for Identifying Outliers in PLCs. In Proceedings of the 2019 56th ACM/IEEE Design Automation Conference (DAC), Las Vegas, NV, USA, 2–6 June 2019; pp. 1–2.
97. Jicha, A.; Patton, M.; Chen, H. SCADA honeypots: An in-depth analysis of Conpot. In Proceedings of the 2016 IEEE conference on intelligence and security informatics (ISI), Tucson, AZ, USA, 28–30 September 2016; pp. 196–198.
98. Buza, D.I.; Juhász, F.; Miru, G.; Félegyházi, M.; Holczer, T. CryPLH: Protecting smart energy systems from targeted attacks with a PLC honeypot. In Proceedings of the International Workshop on Smart Grid Security, SmartGridSec 2014, Munich, Germany, 26 February 2014; Springer: Cham, Switzerland, 2014; pp. 181–192.
99. Holczer, T.; Félegyházi, M.; Buttyán, L. The design and implementation of a PLC honeypot for detecting cyber attacks against industrial control systems. In Proceedings of the International Conference on Computer Security in a Nuclear World: Expert Discussion and Exchange, Vienna, Austria, 1–5 June 2015.
100. Lau, S.; Klick, J.; Arndt, S.; Roth, V. POSTER: Towards highly interactive honeypots for industrial control systems. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, 24–28 October 2016; pp. 1823–1825.
101. Xiao, F.; Chen, E.; Xu, Q. S7commTrace: A High Interactive Honeypot for Industrial Control System Based on S7 Protocol. In Proceedings of the International Conference on Information and Communications Security, ICICS 2017, Beijing, China, 6–8 December 2017; Springer: Cham, Switzerland, 2017; pp. 412–423.
102. 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. 1–6.

103. Wang, Y.; Liu, J.; Yang, C.; Zhou, L.; Li, S.; Xu, Z. Access Control Attacks on PLC Vulnerabilities. *J. Comput. Commun.* **2018**, *6*, 311–325. [CrossRef]
104. Son, J.; Noh, S.; Choi, J.; Yoon, H. A practical challenge-response authentication mechanism for a Programmable Logic Controller control system with one-time password in nuclear power plants. *Nucl. Eng. Technol.* **2019**, *51*, 1791–1798. [CrossRef]
105. Fauri, D.; de Wijs, B.; den Hartog, J.; Costante, E.; Zambon, E.; Etalle, S. Encryption in ICS networks: A blessing or a curse? In Proceedings of the 2017 IEEE International Conference on Smart Grid Communications (SmartGridComm), Dresden, Germany, 23–27 October 2017; pp. 289–294.
106. Yılmaz, E.N.; Gönen, S. Attack detection/prevention system against cyber attack in industrial control systems. *Comput. Secur.* **2018**, *77*, 94–105. [CrossRef]
107. Kleinman, A.; Wool, A. Accurate modeling of the siemens s7 scada protocol for intrusion detection and digital forensics. *J. Digit. Forensics Secur. Law JDFSL* **2014**, *9*, 37. [CrossRef]
108. Yau, K.; Chow, K.P.; Yiu, S.M. A Forensic Logging System for Siemens Programmable Logic Controllers. In Proceedings of the IFIP International Conference on Digital Forensics, DigitalForensics 2018, New Delhi, India, 3–5 January 2018; Springer: Cham, Switzerland, 2018; pp. 331–349.
109. Denton, G.; Karpisek, F.; Breitinger, F.; Baggili, I. Leveraging the SRTP protocol for over-the-network memory acquisition of a GE Fanuc Series 90-30. *Digit. Investig.* **2017**, *22*, S26–S38. [CrossRef]
110. Wu, T.; Nurse, J.R.C. Exploring the use of PLC debugging tools for digital forensic investigations on SCADA systems. *J. Digit. Forensics Secur. Law* **2015**, *10*, 7. [CrossRef]
111. Yau, K.; Chow, K.P. PLC forensics based on control program logic change detection. *J. Digit. Forensics Secur. Law* **2015**, *10*, 5. [CrossRef]
112. Choi, J.; Kim, H.; Choi, S.; Yun, J.H.; Min, B.G.; Kim, H. Vendor-Independent Monitoring on Programmable Logic Controller Status for ICS Security Log Management. In Proceedings of the ACM Asia Conference on Computer and Communications Security, Auckland, New Zealand, 9–12 July 2019; pp. 682–684.
113. Chan, R.; Chow, K.P. Forensic analysis of a Siemens programmable logic controller. In Proceedings of the International Conference on Critical Infrastructure Protection, ICCIP 2016, Arlington, VA, USA, 14–16 March 2016; Springer: Cham, Switzerland, 2016; pp. 117–130.
114. Chan, C.F.; Chow, K.P.; Yiu, S.M.; Yau, K. Enhancing the security and forensic capabilities of programmable logic controllers. In Proceedings of the IFIP International Conference on Digital Forensics, DigitalForensics 2018, New Delhi, India, 3–5 January 2018; Springer: Cham, Switzerland, 2018; pp. 351–367.