SoK: Security of Programmable Logic Controllers

Efrén López Morales
Texas A&M University-Corpus Christi

Ulysse Planta
CISPA Helmholtz Center for Information Security

Carlos Rubio-Medrano
Texas A&M University-Corpus Christi

Ali Abbasi
CISPA Helmholtz Center for Information Security

Alvaro A. Cardenas
University of California, Santa Cruz

Abstract

Billions of people rely on essential utility and manufacturing infrastructures such as water treatment plants, energy management, and food production. Our dependence on reliable infrastructures makes them valuable targets for cyberattacks. One of the prime targets for adversaries attacking physical infrastructures are Programmable Logic Controllers (PLCs) because they connect the cyber and physical worlds. In this study, we conduct the first comprehensive systematization of knowledge that explores the security of PLCs: We present an in-depth analysis of PLC attacks and defenses and discover trends in the security of PLCs from the last 17 years of research. We introduce a novel threat taxonomy for PLCs and Industrial Control Systems (ICS). Finally, we identify and point out research gaps that if left ignored could lead to new catastrophic attacks against critical infrastructures.

1 Introduction

Programmable Logic Controllers, or PLCs, are small rugged computers that have programmable memory to store functions such as timers and logic gates. These functions can control *physical* machines, such as water pumps or centrifuges [36]. PLCs are rather unique systems due to two main characteristics: First, they act as a bridge that connects the cyber and physical worlds; therefore any attack carried out on them can have an immediate effect in the real world. Second, they have opaque, heterogeneous architectures that are different from traditional computer architectures, comprising multiple firmware, which makes them difficult to secure [182].

Before the introduction of PLCs, physical processes were controlled by *relay* panels. Relays are switched either *on* or *off* by an electric current, and thus they can be used in logic circuits. Ladder logic was created as a way to configure relay panels to automatically launch actuation commands based on sensor information. As industrial processes became more complex and varied, relays were not enough to meet the new requirements. An evolution of PLCs, first based on microprocessors, then on networks, and more recently on modern technology paradigms, such as virtualization, has introduced multiple benefits, but at the same time increased the attack surface of these systems.

The first PLC, the *Modicon 084*, was introduced in 1968 [38]. Unlike relays, PLCs could be programmed and reprogrammed to adjust to the process requirements without any physical changes to the control system. A flurry of programming languages emerged, extending ladder logic and introducing new paradigms. Eventually, in 1993, the IEC 61131-3 standard unified these multiple languages into basic standards for PLC programming languages [188].

The first PLCs communicated with the physical world as well as with other computers via analog or serial communications. As computer networks became more reliable and available in the IT world, Ethernet communications were introduced to PLCs. This network connectivity keeps increasing. For example, the Siemens S7-1500 PLC includes up to 4 Ethernet ports, whereas the legacy S7-300 needed an expansion module to support Ethernet connectivity. At the same time, network accessibility now allows remote attackers to deploy classical network attacks against PLCs.

Nowadays, with the advent of Industry 4.0 and IIoT [176], PLCs are going through another paradigm shift bringing even more functionalities like cloud integration, web services, and virtualization. New players like CODESYS and OpenPLC have also entered the market, challenging the long-standing practice of using proprietary hardware and proprietary software products that dominated the PLC industry. These changes pose many open questions about the security of PLCs to the research community.

In order to advance the security of PLCs and the systems they interact with, plenty of research has been produced in the past decades. In particular, research output increased after the term Cyber-Physical System (CPS) was coined in late 2006 by the NSF in the United States [51]. However, the community still lacks an up-to-date general understanding of where the security of PLCs stands and what directions should (or should not) be taken in the future.

To address this challenge, we introduce a comprehensive analysis of the security of PLCs by integrating knowledge from multiple fragmented origins (scientific and grey literature), comprising many existing attack and defense methods in the literature. In addition, we introduce a novel threat model as well as classification and evaluation criteria for summarizing and structuring the existing knowledge about PLC security.

In summary, we make the following contributions:

- We provide a systematization of the literature that consists of **133 papers**, which include **119 attack methods**, and **70 defense methods** (one paper might include one or more attack or defense methods).
- We present a comprehensive taxonomy[1] that integrates the scientific literature with the existing ATT&CK for ICS Matrix in collaboration with MITRE ATT&CK®.
- We identify important PLC security research gaps and discuss future research directions and recommendations.
- We provide **three public tools** to facilitate and foster research and collaboration on this topic. **1)** Our full systematization dataset[2], which other researchers can use to replicate our results and perform their own analyses. **2)** A PLC security artifacts repository[3], which acts as a centralized database for artifacts updated by the community. **3)** A reference graph[4] that enables researchers to further analyze the reviewed literature in this SoK.

2 Background

We now introduce some relevant background topics on PLCs that will be leveraged in future sections.

2.1 PLC System Environment

PLCs [168] are an essential component of most physical critical infrastructures such as water treatment systems and gas pipelines. PLCs are commonly found in Industrial Control Systems and Supervisory Control and Data Acquisition (SCADA) systems. In the last few years, two new concepts have emerged in the context of Cyber-Physical Systems: Industrial IoT (IIoT) and Indystry 4.0. IIoT is a subset of IoT concerned with connecting industrial assets, e.g., PLCs, with information systems and business processes. Industry 4.0, on the other hand, is a subset of IIoT and refers to the use of Internet technologies to improve production efficiency by employing smart services in smart factories [176]. Figure 12 illustrates the relationship between the above concepts. Furthermore, IIoT and Industry 4.0 have changed PLCs in two ways: First, they have introduced support for modern network protocols, which we discuss further in Sec. 2.3. Second, they have changed the hardware and software architectures to support virtualization and compatibility. We further discuss these topics in Sec. 2.4.

[1] https://github.com/efrenlopezm/ics2matrix
[2] https://github.com/efrenlopezm/plc-sok-dataset
[3] https://github.com/efrenlopezm/plcsecurityartifacts
[4] https://www.researchrabbitapp.com/collection/public/E6XRY0186G

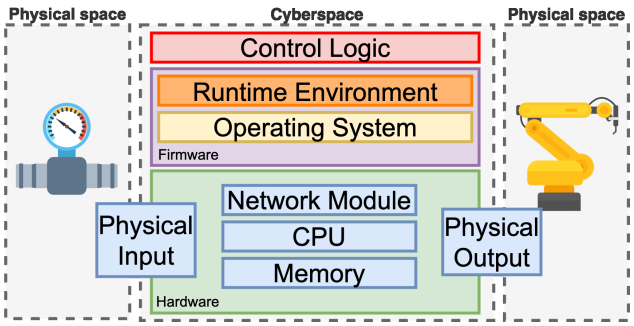

Figure 1: A Generalized PLC Architecture. Based on [36, 69].

PLCs' industrial environment, however, remains largely the same and typically includes the following [183]:

Actuator. A hardware component that moves or operates a device in the physical world. Examples of actuators include valves, motors, and piezoelectric actuators.

Sensor. A device that generates an electrical analog or digital signal that represents a physical property of a process. Examples include temperature or magnetic field sensors.

Engineering Station. A general-purpose computer that is used to write the control logic or ladder logic code for the PLC to execute. It is usually connected to the PLC so that the compiled control logic program can be uploaded.

Human Machine Interface (HMI). The hardware or software used to interact with the PLC, e.g, a physical control panel with buttons and lights or a software display.

We further discuss PLCs' underlying architecture and environment in Appendix C.

2.2 PLC Basic Components

As shown in Fig. 1, a typical PLC has the following basic components that are susceptible to attacks:

Control Logic. A control logic program contains the instructions that the PLC executes to interact with its environment. The IEC 61131-3 standard [188] specifies the syntax and semantics of a unified suite of programming languages for PLCs. Control programs are written in one of these supported languages, e.g., Structured Text, and then compiled into machine code or bytecode. For instance, Siemens PLCs run proprietary MC7 bytecode [144] compiled by the SIMATIC S7 Manager software [173].

Runtime Environment. The runtime environment executes the process control code [69] and interacts with the I/O modules. It can be proprietary, e.g., Schneider, or open source like the OpenPLC runtime [19].

Operating System. Most PLCs have a Real-Time Operating System (RTOS) [181]. RTOS are operating systems that meet strict processing time requirements and support real-time applications. Vendors support a variety of RTOS in their platforms. For example, Siemens supports the Nucleus RTOS [175] and VxWorks [174].

Firmware. The firmware bridges the gap between the PLC hardware and software. While simple PLCs might run applications as bare metal (without an OS) [27,215], modern PLCs use the firmware under an RTOS. Firmware can be upgraded via SD cards or through a network connection [98].

CPU. The CPU interprets the input signals and executes the logic instructions saved in memory. The CPU chassis has slots where other components may be attached, e.g., a network module. It may also include USB ports and SD Card slots.

Memory Unit. It stores the program that the CPU will execute along with input data. The memory unit may include different types of memory blocks, which are further discussed in Appendix D.

Network Module. Modern PLCs can have one or more ports to communicate with the supervisory control network (regular computers monitoring the process) or the fieldbus (actuators and sensors).

Physical I/O Modules. These include input modules with metal pins that receive information (via a voltage or current analog signal) from sensors. The output modules send analog data to actuators such as servo motors.

PLC Scan Cycle. It is the cycle in which the PLC reads the sensor inputs, executes the current control logic program and updates the output to the actuators. It is measured in ms and should stay constant. If its time increases, PLCs implement a watchdog timer that sends the PLC to a *Fault* mode [8, 24].

2.3 PLC Communication Protocols

As discussed in the introduction, PLCs are becoming more connected via modern communication protocols. Many of these protocols were not designed to include strong security features [101], allowing for the proliferation of vulnerabilities that make the PLCs running them susceptible to attacks [34, 208]. Protocols commonly used in practice include, but are not limited to, the following:

Fieldbus. PLCs use Fieldbus [187] protocols to talk to sensors and actuators. Historically these communications were done through serial-based interfaces or analog signals. Sample standards include Profibus, CAN bus, Modbus, and DeviceNet. They all differ in features and implementation, resulting in limited compatibility [67, 112].

Supervisory Network Protocols. PLCs communicate with other controllers or classical computers through a variety of proprietary and standardized protocols. These protocols use the IEEE 802.3 Ethernet standard [100] in industrial environments [127]. For example, the EtherNet/IP protocol combines IEEE 802.3 and the TCP/IP Suite [62]. Some are designed to operate on LANs (they do not use IP addresses but only MAC addresses), such as GOOSE, while others use TCP/IP and can be used on LANs and WANs, such as Modbus TCP.

Industry 4.0 Protocols. These protocols include support for cloud connectivity, compatibility between devices, and security features. Examples include MQTT (Message Queuing Telemetry Transport) [140] and OPC-UA (Open Platform Communications-Unified Architecture) [194]. Currently, some PLC models support Industry 4.0 protocols [44], whereas older PLCs can be retrofitted to support them [92].

2.4 SoftPLCs

The term SoftPLC has been used in the scientific literature since the late 1990s [54, 159, 179]. Although it has not been well defined, generally a SoftPLC is a software runtime environment that executes PLC programs, for example, programs that follow the IEC 61131-3 standard [188]. The runtime is portable and compatible with multiple hardware that can range from microcontrollers to cloud servers [147]. The two major SoftPLC projects are CODESYS [47] and OpenPLC [147]. CODESYS supports approximately a thousand different device types from more than 500 manufacturers [46]. OpenPLC, first proposed in 2014 [22], includes a runtime and an editor, and it is compatible with 18 platforms including Windows and Linux [147]. One of the biggest differences between OpenPLC and CODESYS is that OpenPLC's runtime is open source [19], unlike that of CODESYS. For the remainder of this work, we will refer to traditional PLCs as *HardPLCs*, in order to differentiate them from SoftPLCs.

3 Research Questions and Methodology

In this section, we first elaborate on the research questions addressed in this paper, and we then describe the methodology we followed to systematize knowledge by collecting, analyzing, and evaluating works in the literature.

3.1 Overview of Research Questions

RQ-1: ***What are the attack methods against PLCs?*** We aim to categorize and analyze attack methodologies targeting PLCs introduced in the literature in the last 17 years.

- **RQ-1.1:** *Which components of the PLC are targeted?* We aim to identify what internal components of a PLC, e.g., the CPU described in Sec. 2.2, are being targeted, in an effort to identify patterns such as the components that have received the most attention in the literature.
- **RQ-1.2:** *How difficult is it to deploy attacks?* We also aim to identify the level of effort required by attackers to successfully carry out attacks against PLCs. In Sec. 4, we elaborate on a threat model and classification criteria, including the

potential attack vectors as well as the level of access required by attackers, e.g., Internet access, needed to deploy an attack.

- **RQ-1.3:** *What is the impact of deploying attacks?* Finally, we are interested in identifying the impact that attacks against PLCs may have if they are deployed successfully. This includes the level of disruption achieved, e.g., modifications to control logic programs as shown in Sec. 2.1.

RQ-2: *What are the defense methods to protect PLCs?*
In addition to attacks, we want to systematize the PLC defenses proposed in the literature in the last 17 years.

- **RQ-2.1:** *Which components of the PLC are protected?* We want to identify which of the components introduced in Sec. 2.2 are the focus of defenses to understand which of them have received less attention and may therefore remain unprotected.
- **RQ-2.2:** *How difficult is it to deploy defenses for PLCs?* We also study the level of effort required to deploy defense mechanisms for PLCs. As shown in Sec. 4.5, this includes the organizational effort from administrators and/or operators to modify and adjust a given PLC/ICS Environment and the performance overhead.
- **RQ-2.3:** *Are there enough defenses addressing reported attacks for PLCs?* Finally, we are interested to know if enough defenses exist in the literature to counteract the attacks we have identified as a part of this work. This way, we aim to identify research gaps and discuss the directions of future work as we describe in Sec. 6 and Sec. 7.

3.2 Knowledge Systematization Methodology

To answer the Research Questions raised in Sec 3.1, we use the systematization methodology shown in Fig. 2. First, we perform a systematic literature review. Second, based on the data obtained in the literature review, we perform data analysis and modeling. Finally, we assess the data.

❶ **Systematic Literature Review**. We review both scientific and grey literature to collect PLC attack and defense methods. We also define the final SoK scope based on the specified inclusion criteria.

Scientific Literature Review. We consider scientific literature to mean the literature that is based on the scientific method that uses evidence to develop conclusions. It uses previous literature to develop theories and hypotheses while taking care to cite the authors and tools that are used. For the scientific literature review, we carried out the following steps: **1)** Search Querying. We used search queries based on keywords such as *"plc"*, *"ics"*, and *"security"*. The complete list of keywords can be found in Appendix A. **2)** Consulting literature resources. We used these queries to search the selected resources such as Google Scholar and the ACM Digital Library. The complete list of consulted resources can be found

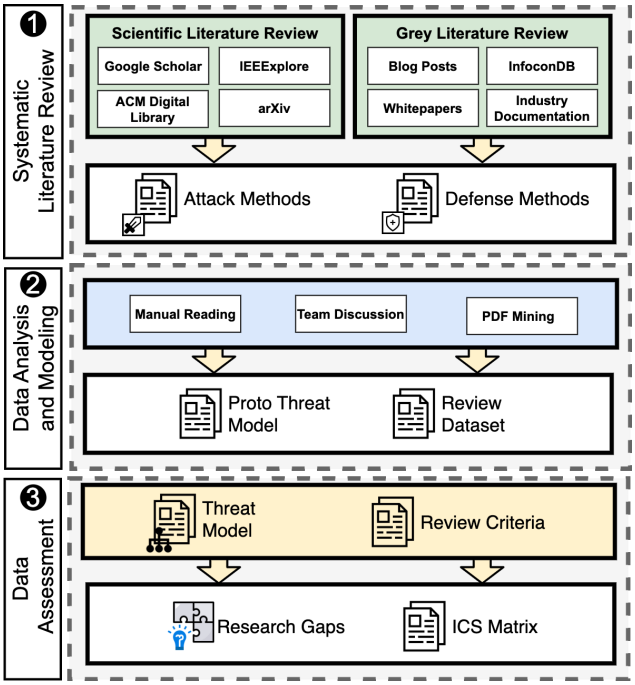

Figure 2: Our systematization methodology. Based on [116].

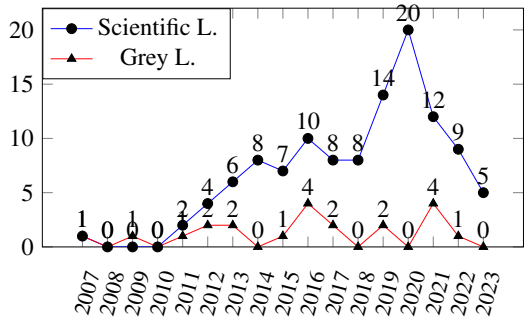

Figure 3: Literature focused on PLC Security per Year.

in Appendix A. **3)** Applying inclusion criteria. After collecting all these papers, we include them only if they meet the following three criteria. A) The paper must include the term "PLC" or "Programmable Logic Controller". B) It proposes an attack or defense method. C) It includes an experimental evaluation of an attack or defense with at least one PLC. **4)** Applying snowballing. We applied the snowballing technique [206] on all the papers that matched the above criteria to find papers that went unnoticed during the initial search.

Grey Literature Review. We consider grey literature as literature with limited distribution (i.e., not included in academic publishing libraries). It includes unpublished reports, policy documents, white papers, and technical reports [48]. To perform our grey literature review, we followed the same steps used in the scientific literature review, except that in step 2) (consulting literature resources) we instead searched in the grey literature resources listed in Appendix A.

Final SoK Scope. As a result of the scientific and grey literature reviews we collected 133 papers with the earliest being from 2007 and the latest from 2023. Although we searched for papers before 2007, we did not find any that met our criteria. To the best of our knowledge, and based on our exhaustive research, there was no research on designing new attacks or defenses for PLCs before 2007. Fig. 3 shows the final number of scientific and grey papers included in our selection criteria per year of publication, depicting an overall increase in PLC security research over the years, peaking in 2020.

❷ **Data Analysis and Modeling**. In this phase, we first read and analyzed each paper to extract important security-relevant information such as attack vectors, PLC models and manufacturers, and PLC target component. Second, we matched each of the attack and defense methods to their corresponding MITRE technique, subtechnique or mitigation category. The MITRE framework is further discussed in Sec. 4.1. Third, we recorded this information in a spreadsheet. The result of this data analysis and modeling was the identification of the building blocks for a first draft of our threat model depicted in Fig. 4 and the classification criteria discussed in Sec. 4., e.g., the access level and PLC target component.

❸ **Model Assessment**. In this phase, we evaluate the data, threat model, and criteria developed in ❷ to produce the SoK contributions. We use these results to summarize our findings and identify the research gaps discussed in Sec. 6. Finally, we evaluate the results to produce the ICS threat taxonomy discussed in Sec. 4.1.

4 Classification and Evaluation Criteria

In this section, we present the criteria to classify and evaluate the works considered for this SoK. These criteria and the symbols introduced alongside them will later be used in Tables 1 and 2. Our first research question RQ-1 is addressed in Sec. 4.2, Sec. 4.3, and Sec. 4.4. Our second research question RQ-2 is addressed in Sec. 4.5, Sec. 4.6 , and Sec. 4.7.

4.1 ICS Threat Taxonomy

To better classify attack and defense methods in the following sections, we extended the MITRE ATT&CK for ICS Matrix [128] and the Hybrid ATT&CK Matrix [11] to create the ICS^2 Matrix, a taxonomy of threats against PLCs and ICS. The taxonomy includes adversary tactics that describe "what" is the adversary's goal and attack techniques that describe "how" the adversary can complete their goal. Additionally, it includes mitigations that prevent a technique from being successfully executed. The ICS^2 Matrix incorporates the scientific knowledge accumulated during the past 17 years of PLC security research by adding 6 new attack techniques and 5 new mitigation categories based on the literature reviewed in this SoK. Due to space restrictions, we provide a condensed version of our taxonomy in Fig. 13. The full ICS^2 Matrix version is publicly available[5] for the security community to use and extend. Additionally, at the time of writing this paper, we are actively working with MITRE to integrate our research findings within the next version of MITRE ATT&CK for ICS [128] and D3FEND Knowledge Graph [106, 138].

Matrix Integration. One of the main limitations of the MITRE ATT&CK for ICS Matrix [128] is that it does not incorporate techniques based on security research findings but instead focuses on techniques based on real-world ICS cyber-attacks. For example, the "system firmware" technique [186] includes a reference to Triton discussed in Appendix F. However, it does not include any references to any of the PLC Firmware research listed in Table 2. This was also confirmed by MITRE during one of our communications. Our ICS^2 Matrix fills this gap. To integrate our systematization findings with the existing MITRE for ICS Matrix, we needed to add techniques not already covered by MITRE. To develop these new techniques, we used MITRE's official guidelines [137]. This document outlines the information required to create a new MITRE-style technique, for example, a matching "tactic" and "procedure example". In Appendix B we describe in detail how we developed and included one of these techniques.

4.2 Attack Classification

Considering RQ-1.1, the following criteria classify attacks according to the taxonomy technique, e.g., Adversary-in-the-Middle, the basic components that are targeted, e.g., the CPU or Memory, and the attack vectors that can be ultimately used against a PLC.

Target PLC Component. Each attack method was classified according to the PLC Basic Component that is the primary target, following the description previously shown in Sec. 2.2. Such information was retrieved based on the descriptions explicitly provided by the paper's authors.

Attack Category. Each attack method was sorted into one *technique* category, based on our ICS^2 Matrix. The technique categories allowed us to determine what kind of defenses might (or might not) counter them.

Attack Vector. When launching an attack against a PLC, there might be one or more vectors or paths available to the adversary to deliver the payload. The list of attack vectors we considered are shown in Table 4. Most of the time the attack vector is a vulnerability in the implementation of a network protocol, e.g., S7comm. However, there are other vectors like inserting an SD card into the PLC chassis, for instance, the work by Garcia et al. [75].

4.3 Attack Complexity

Following RQ-1.2, the criteria shown next are intended to understand, qualify, and quantify how complex it is for an

[5] https://github.com/efrenlopezm/ics2matrix

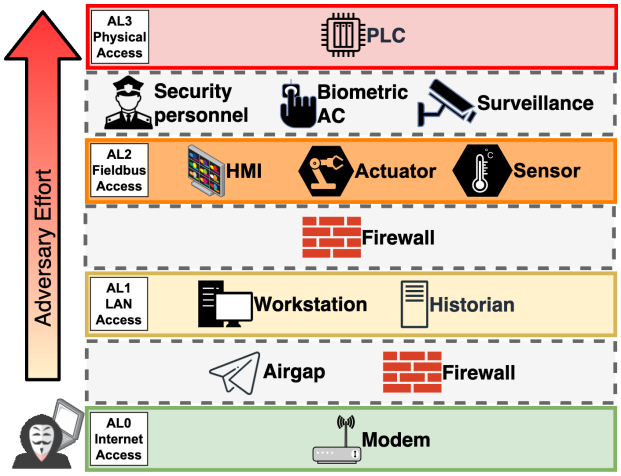

Figure 4: PLC-centered Threat Model. Based on [183].

attacker to carry out an attack against a target PLC.

Environment Knowledge. This criterion evaluates how much knowledge of the cyber-physical system environment where the PLC lives, e.g., the system topology discussed in Sec. 2.1, is required to launch the attack. An attack may require *zero knowledge* (○), e.g., a DoS [212] attack targeting the PLC network module only requires an IP address and does not require any environment knowledge. *Partial knowledge* (◑) may require basic information only, e.g., a high-level description of the physical process. CaFDI [132], for example, requires limited local information about the substation configuration. Finally, *extensive knowledge* (●) requires detailed information about the cyber-physical environment. For example, SABOT [131] requires the adversary to encode their understanding of the system behavior into a specification.

Available Source Code. This criterion evaluates whether the attack source code is publicly available (●) or not (○).

Access Level. We also consider the level of access an attacker needs to carry out the suggested attacks. Our threat model, shown in Fig. 4, is divided into four distinct *Access Levels* (ALs) labeled from AL0 to AL3. Adversaries might try to move through the ICS environment in order to reach the PLC and deliver their exploit payload. Overall, our threat model helps us answer two main questions: *1) What access level is required to attempt the attack?*, and *2) What attack surfaces are available to the adversary?*

AL0: Access to PLC via the Internet (0). An adversary may be able to access a PLC through an Internet-compatible protocol, e.g., S7comm. This is usually a private network where a remote SCADA computer connects to a PLC via an industrial protocol such as DNP3. Attacks may also be launched if a PLC is publicly exposed on the Internet. At first, this might seem like a ludicrous idea: Why would anybody make a PLC publicly available over the Internet? However, a preliminary Shodan search suggests that the number of publicly accessible PLCs (excluding Shodan-identified honeypots) is over 6,500 as of September 2023[6]

AL1: Access to Supervisory LAN (1). This level requires an attacker to have access to the LAN network of the industrial process. This can be achieved if the attacker compromises workstation computers, data historians, or similar operational computers. Some industrial protocols run only on Ethernet and are non-routable (such as GOOSE), so an attacker targeting a vulnerable GOOSE stack on a PLC will have to do so in the same LAN.

AL2: Access to PLC Fieldbus Network (2). As mentioned in Sec. 2.3, fieldbus represents the lower level of communications between the PLC and nearby field equipment such as actuators and sensors. An attacker can compromise any of these devices to exploit a vulnerability in the fieldbus code implementation. The attacker can also launch Adversary-in-the-Middle attacks in this network, as fieldbus communications are rarely authenticated in practice.

AL3: Physical Access to PLC (3). Lastly, this level assumes the adversary has bypassed environmental and physical protection measures (e.g., locked doors) of the target. Having physical access to the PLC may be the most difficult scenario for an adversary but can still be achieved by malicious insiders or disgruntled employees. An attacker with physical access can use a JTAG interface, as shown in Table 4.

4.4 Attack Impact

Following RQ-1.3, the criteria below are intended to understand and quantify the potential impact of a successfully carried out attack, e.g., the *payoff* in return of the effort invested, as discussed in Sec. 4.3.

Potential Damage. This criterion evaluates the *immediate* damage inflicted to the physical operation of the ICS. The damage is *limited* (1) if the attack does not change the operation of the process but instead aims to collect information or gain unauthorized access, for example, a password sniffing attack [202]. The potential damage is *substantial* (2) if the attack can stop the industrial process partially or completely, e.g., a DoS attack [146]. The potential damage is *severe* (3) if the attack is able to insidiously control the physical process, e.g., a logic bomb or firmware modification attack, where the attacker can launch arbitrary control commands to the system (similar to Stuxnet, discussed in Appendix F).

HardPLC Targets. An attack may affect a *single* PLC model (○), e.g., the Siemens SIMATIC S7-300, or may affect *multiple* PLC models of the same manufacturer (◑), e.g., the Rockwell's Allen-Bradley 1100 and 1400 models. Also, an attack may affect *multiple* PLC models from two or more manufacturers (●), e.g., Siemens' SIMATIC S7-1500 and Modicon's M221. An attack that is effective against various devices is an indicator of the severity of the problem, while an attack focusing on a single device might be contained.

[6] https://www.shodan.io/search?query=plc+-%22792%2F71644%22.

SoftPLC Targets. Finally, this criterion indicates if a SoftPLC was used to evaluate the proposed attack. An attack may have been evaluated using *CODESYS* (C), *OpenPLC* (O) or *no SoftPLC* at all (-) (i.e., the attack only focused on HardPLCs).

4.5 Defense Classification

After describing our criteria for attacks, we now focus on defenses by looking at RQ-2.1.

Defense Category. Each defense method was sorted into one *mitigation* category, based on our ICS^2 Matrix. The mitigation categories allowed us to determine what kind of attacks might (or might not) be countered.

Defense Vector. The defense vector is the path used in order to stop the attack payload from being delivered. The available defense vectors are defined in our Threat Model for PLCs in Table 4. Most of the time, the attack and defense vectors will match. However, they can differ. For example, *Rajput et al.* use the JTAG interface as a defense vector [156].

4.6 Defense Deployability

Following RQ-2.2, the following criteria evaluate how difficult it is to deploy a defense method to protect PLCs.

PLC Overhead. This criterion quantifies the overhead that the defense method incurs on the PLC itself, which can be either *zero* (1), *negligible* (2), or *considerable* (3).

Infrastructure Overhead. This criterion quantifies the cost that comes with setting up the required infrastructure to implement a defense method. That may involve *no changes* (1), or it may involve *either* an infrastructure change, e.g., setting up new VLANs or an additional hardware component, e.g., bump-in-the-wire (2). Finally, a defense may involve *both* infrastructure changes and hardware components (3).

Maintenance. This criterion quantifies the level of post-deployment maintenance required by a given defense method. A defense method may require *no maintenance* (1), *sporadic maintenance* (2), or *constant maintenance* (3).

Source Code Availability. This criterion evaluates whether the defense source code is publicly available (●) or not (○).

4.7 Defense Robustness

Following RQ-2.3, we also characterize how the defense mitigates attacks.

Defense Stage. Defenses can be categorized in three different stages of the security process. 1) *Prevention Defense* (PR) aims to reduce the possibility of an incident *before* a known vulnerability is exploited. 2) *Detection Defense* (DE) aims to identify and alert about a *current or recent* attack. 3) *Recovery Defense* (RE) aims to reduce the damage of an attack *after* it has been carried out successfully.

Effectiveness. The defense method's effectiveness is indicated by its accuracy. The accuracy values are taken from the research paper itself. If the paper did not report or specify the accuracy, it is marked as *N/S* (not specified). We selected accuracy because it is the most reported metric across the defense method papers. We recognize that this metric does not work for all defense methods, however, it is the most frequently reported quantitative metric available in the current literature. We further discuss this problem in Sec.7.

HardPLC Scope. This criterion evaluates the number of different PLC models that are protected by the defense. The defense may protect a *single* PLC model (○), e.g., the Siemens SIMATIC S7-300, or it may protect *multiple* PLC models of the same manufacturer (◑), e.g., the Rockwell's Allen-Bradley 1100 and 1400 models. Also, a defense may protect *multiple* PLC models from two or more manufacturers (●), e.g., Siemens' SIMATIC S7-1500 and Modicon's M221.

SoftPLC Scope. This criterion indicates if any SoftPLC was used to evaluate a defense, either *CODESYS* (C), *OpenPLC* (O) or *no SoftPLC* at all (-).

5 Overview of Attacks

By using the systematization methodology presented in Sec. 3 and the criteria discussed in Sec. 4, we obtained Tables 1, and 2. These tables use the *PLC target component* categorization defined in Fig. 1.

We now illustrate examples of attacks per target component. Due to the large number of papers in our SoK (133 papers), we cannot discuss all of them. To select the papers systematically, we picked the two papers with the most citations from each category. We used Google Scholar to perform the search on June 3, 2023. In the next paragraphs, we discuss the insights provided by these highly-cited papers.

Attacks that Target the PLC Communications Module. *Urbina et al.* [195] introduce an AitM attack between the PLC and the field devices (AL2). An interesting observation of that paper is that field networks tend to follow a ring topology rather than the typical star topology of switched Ethernet networks; therefore, an AitM attack does not need to use ARP (Address Resolution Protocol) spoofing or similar techniques to place itself between two devices. It just needs to inject the attack between the two attacked devices. An AitM attacker can then send false sensor readings to the PLC or false control commands to actuators.

Wardak et al. [202] introduce another network attack, which focuses on password sniffing on the network interface between a workstation and the PLC (AL1) in Fig. 4. The authors show that several of the connections between the workstations and the PLCs are not encrypted (or authenticated). Consequently, passwords can be sniffed and then used to gain access to protected actions in the PLC (e.g., the attacker can send start and stop commands to the PLC).

Attacks that Target the Control Logic. Two of the most important challenges for modifying the control logic of a PLC are 1) how to infect the PLC without being detected and

Attacks												Defenses											
				Complexity			Impact							Deployability				Robustness					
Target Component	Method	Category (Technique)	Attack Vector	Access Level	Env Knowledge	Source Code	P. Damage	HPLC Targets	SPLC Targets			Method	Category (Mitigation)	Defense Vector	PLC Overhead	Infra Overhead	Maintenance	Source Code	Effectiveness	D. Strategy	HPLC Scope	SPLC Scope	
Network	Attacks on situational awareness [115]	Adversary-in-the-Middle	P	1	●	○	2	○	-			Cross-layer fingerprinting [71]	Network Intrusion Pre.	D	1	1	1	○	92.8	DE	○	-	
	Concealment Attack [61]		M	1	○	●	2	○	-			DFA-based intrusion detection [83]		M	1	2	1	○	99.8	DE	○	-	
	Controller Eavesdropping [78]		P	1	○	○	1	●	O			Encrypted Traffic IDS [95]		I	1	2	1	○	N/S	DE	-	-	
	Controller Packet Tampering [78]		P	1	○	○	3	●	O			IDS for S7 networks [110]		S7	1	2	1	○	99.8	DE	○	-	
	False Data Injection Attack [43]		M	0	○	○	2	●	-			Model-based anomaly detection [113]		S7	1	2	1	○	96	DE	○	-	
	Man in the middle [25]		E	1	○	○	1	○	-			Physical fingerprinting [71]		D	1	1	1	○	92.8	DE	○	-	
	OPC-UA Rogue Client [10]		UA	0	○	○	2	○	-			PLC Watermarking [8]		E	1	2	1	○	N/S	DE	●	-	
	Replay attack [202]		P	0	○	○	2	○	-			Embedding Encryption [20]	Encrypt Network Traffic	M	1	1	1	○	N/S	PR	-	O	
	Replay attack [165]		F	0	○	○	2	○	-			Hash Authentication [74]		F	1	1	1	○	N/S	PR	○	-	
	SDN Enabled MitM [78]		P	1	○	○	1	●	O			LVST (LabView SSH Tunnel) [152]		E	1	2	1	○	N/S	PR	●	-	
	ISO-TSAP Replay Attack [31]		P	0	○	○	2	◑	-			AES-256 Implementation [21]		M	1	1	1	●	N/S	PR	-	O	
	Third-party Eavesdropping [78]		P	1	○	○	1	●	O			PLCrypto [209]		E	1	1	1	●	N/S	PR	○	-	
	Change the IP [161]	Denial of Service	E	0	○	●	2	○	-			PLC-Sleuth [210]		S7	1	2	1	○	100	DE	○	-	
	Control engine attack [154]		M	0	○	○	2	●	-			Semantic IDS [89]	Network Intrusion Pre.	M	1	2	1	○	N/S	DE	-	-	
	Denial of Service Attack [212]		P	0	○	○	2	○	-			Telemetry IDS [153]		M	1	2	1	○	99.5	DE	-	-	
	DoS clearing Flow Table [78]		P	0	○	○	2	○	O			Zeus [90]		E	1	2	2	○	98.9	DE	○	-	
	Flow Rule Blocking [78]		P	0	○	○	2	○	O			Side-channel Anomaly Detection [58]	Out-of-Band Comms Channel	E	1	1	2	○	N/S	DE	○	-	
	UDP reflect attack [165]		F	0	○	○	2	○	-			Anomaly detection [219]		N	1	2	2	○	90	DE	●	-	
	Authentication Bypass [87]	Modify Auth. Process	E	1	○	○	1	○	-			Arcade.PLC [33]	Control Logic Verification	S7	1	1	1	○	N/S	PR	○	-	
	Cryptographic attack [165]		F	1	○	○	1	○	-			ShadowPLCs [104]	Network Intrusion Pre.	S7	2	1	2	○	97.3	DE	○	-	
	Replay attack [120]		S7	0	○	○	1	◑	-			Snapshooter [102]		S	2	2	1	○	N/S	DE	-	O	
	S7 Authentication Bypass [31]		P	0	○	○	1	◑	-			Traffic Data Classification [117]		S7	1	1	1	○	99.7	PR	○	-	
	Unauthorized password updating [202]		S7	0	○	○	1	○	-			vBump [189]		G	1	3	2	●	N/S	PR	○	C	
	SoMachine Authentication [25]	Hardcoded Credentials	M	0	○	○	1	○	-														
	Credentials from storage [25]		E	0	○	○	1	◑	-														
	subverting read/write-protection [25]		P	0	○	○	1	○	-														
	subverting write-protection [25]		P	0	○	○	1	○	-														
	Command packet flooding [86]	Network Denial of Service	E	0	○	○	1	○	-			Modbus/TCP Firewall [171]	Filter Net. Traffic	M	1	2	1	○	N/S	PR	○	-	
	Modbus Flooding Attack [32]		M	0	○	○	2	○	-														
	Network Flooding Attack [146]		M	0	○	○	2	●	-														
	UDP flooding Attack [109]		M	0	○	○	2	○	-														
	Password reset attack [25]	Brute Force	M	0	○	○	1	○	-			Shade [214]	Network Intrusion Pre.	E	2	1	1	●	N/S	DE	●	-	
	CLIK password attack [105]		M	0	○	●	1	○	-			Host Anomaly Detecton [69]		M	2	1	1	○	N/S	DE	●	-	
	Dictionary Attack [37]		B	0	○	○	1	○	-			Argus [5]		E	1	3	3	○	N/S	DE	-	-	
	Dump module code [161]	Data from Local System	E	0	○	●	1	○	-			PLC-PROV [9]	Validate Program Inputs	N	1	2	2	○	N/S	DE	-	C	
	Memory Logic Attack [31]		P	0	○	●	1	○	-			PLCPrint [50]		S7	1	1	1	○	95	DE	●	-	
	Crash 1756-ENBT module [161]	Device Restart/Shutdown	E	0	○	●	2	○	-			ABAC model for PLC [85]	Authorization Enforcement	S7	1	2	1	○	N/S	PR	●	-	
	Reset 1756-ENBT module [161]		E	0	○	●	2	○	-			FINS detection rules [73]	Network Allowlists	F	1	2	1	○	N/S	PR	◑	-	
	Leak Modbus data [193]	Exfiltration Side-channel	M	0	○	○	1	○	C														
	Leak OPC-UA Process Data [193]		UA	0	○	○	1	○	C														
	Passive network scanner [203]	Network Conn. Enumeration	M	1	○	○	1	●	-														
	Port scanner [111]		P	0	○	●	1	○	C														
	SOCKS Proxy [111]	Connection Proxy	P	0	○	○	1	○	C			SDN-enabled automatic response [142]	Network Intrusion Pre.	E	1	2	1	●	N/S	DE	-	-	
	New ADMIN account [37]	Exp. for Credential Access	B	0	○	○	1	○	-			WeaselBoard [141]	Exploit Protection	BP	3	3	3	○	N/S	PR	●	-	
	Getting a Shell on the PLC [31]	Exp. for Privilege Escalation	P	0	○	○	3	◑	-			Memory Access Taintedness [29]	Exploit Protection	M	-	1	1	○	N/S	DE	-	-	
	Password stealing [202]	Network Sniffing	P	0	○	○	1	○	-			SCADA Protocol Obfuscation [28]	Encrypt Net. Traffic	M	1	1	1	○	N/S	PR	○	-	
	Wireless Fieldbus MiTM [195]	Wireless Sniffing	E	3	○	○	1	○	-			Secure Logging for ICS [166]	Encrypt Net. Traffic	S	1	3	1	○	N/S	PR	○	-	
OS	3S CoDeSys Tools [204]	Exp. for Privilege Escalation	C	1	○	●	2	-	C														
	Remote arbitrary code execution [108]		P	0	○	○	2	◑	-			ECFI [3]	Exploit Protection	RT	2	2	1	○	N/S	PR	○	O	
	Arbitrary Code Execution [4]	Exp. for Client Execution	R	3	○	○	2	○	-														
Runtime	Ghost in the PLC [2]	Block I/O Communication	R	3	◑	○	3	●	C			GhostBuster [1]	Exploit Protection	RT	2	1	1	○	N/S	DE	○	C	
	Unauthenticated file read/write [82]	Data from Local System	C	1	○	○	2	○	C														
	XML bomb [82]	Modify Parameter	C	1	○	○	2	○	C														

Table 1: A Summary of PLC Attack and Defense Methods (Network, OS and Runtime Components).
D=DNP3, **G**=GOOSE, **S7**=S7COMM, **I**=IEC 104, **UA**=OPC-UA, **P**=Profinet, **E**=EtherNet/IP, **M**=Modbus TCP, **F**=FINS, **S**=Syslog, **B**=Beckhoff, **BP**=Backpane, **R**=SoC Register, **RT**=Runtime, **O**=OpenPLC, **C**=CODESYS / CODESYS upload protocol, **N**=Not Specified, **DE**=Detection, **PR**=Prevention, **RE**=Recovery

Attacks											Defenses										
Target Component	Method	Category (Technique)	Attack Vector	Complexity: Access Level	Complexity: Env Knowledge	Complexity: Source Code	Impact: P. Damage	Impact: HPLC Targets	Impact: SPLC Targets		Method	Category (Mitigation)	Defense Vector	Deployability: PLC Overhead	Deployability: Infra Overhead	Deployability: Maintenance	Deployability: Source Code	Robustness: Effectiveness	Robustness: D. Strategy	Robustness: HPLC Scope	Robustness: SPLC Scope
Control Logic	Advanced Stealthy Injection Attack [17]	Modify Control Logic	P	0	◑	●	3	○	-		Safety Verification [77]	Control Logic Verification	S7	2	2	1	○	N/S	DE	○	-
	CLIK [105]		SM	1	○	●	3	○	-		CPLCD [211]		S7	1	1	1	○	N/S	DE	○	-
	Data Execution Attack [213]		M	0	○	●	2	○	-		Trusted Safety Verifier [133]		N	1	2	1	○	N/S	DE	-	-
	Fragmentation and Noise Padding [213]		M	0	○	●	2	●	-		Detection Manipulation [119]		X	1	1	1	○	N/S	DE	○	-
	Ladder Logic Bomb [84]		U	3	●	○	3	○	-		PLCloud [170]		S7	1	3	3	○	N/S	RE	○	-
	Time-of-Day (TOD) interrupt attack [16]		S7	0	◑	○	3	○	-		PLC Guard [125]		S7	1	2	2	○	N/S	PR	○	-
	Control logic injection attack [15]	Modify Program	X	0	◑	○	3	○	O		PLC-VBS [124]	Vulnerability Scanning	S7	1	1	1	○	N/S	PR	○	-
	Dynamic Malware Payloads [129]		N	1	○	○	3	-	-		ICSPatch [143]	Update Control Logic	N	1	2	1	●	100	PR	◑	C
	Executing arbitrary ladder logic [82]		N	0	○	○	2	○	-		D-Box [135]	Limit Access to MCU R.	X	2	1	1	●	N/S	PR	○	-
	Immediate Failure Attack [121]		T	1	◑	○	2	○	-												
	Latent Failure Attack [121]		T	1	●	○	3	○	-												
	SABOT [131]		N	1	●	○	3	-	-												
	Ladder Logic Exfiltration [53]	Exfiltration over Side-channel	U	3	◑	○	1	○	-												
	Leak Application data [193]		N	0	○	○	1	-													
	S7-1200 Download Attack [34]	Program Download	S7	0	○	○	2	◑	-												
	S7-1500 Download Attack [34]		S7	0	○	○	2	○	-												
	Denial of Engineering Operations [169]	Adversary-in-the-Middle	E	1	○	●	2	○	-		Optimised Datablocks [158]	Encrypt Sensitive Info.	S7	2	2	1	○	N/S	PR	○	-
	CaFDI [132]	Brute Force I/O	N	2	◑	○	3	-	-												
	Time-of-Day Attack [18]	Change Operating Mode	S7	1	◑	○	2	○	-		PLC redundancy framework [103]	Redundancy of Service	E	2	3	2	○	N/S	PR	○	-
	LogicLocker [70]	Data Encrypted for Impact	M	0	○	○	2	●	-		Quad-redundant PLC [123]		X	2	3	2	○	N/S	PR	○	-
	Denial of Decompilation Attack [220]	Denial of Service	S7	0	○	○	2	●	-		Digital twin-based simulation [55]		N	1	3	3	●	N/S	RE	-	-
	Stable Perturbation Attack [163]	ICS Sector Discovery	N	1	◑	●	3	○	-												
	Evil PLC Attack [162]	Lateral Tool Transfer	X	0	◑	○	3	●	-												
	Targeted Manipulation of FB Operation [88]	Manipulate I/O Image	N	1	○	○	3	●	-												
Firmware	Bricking the device [82]	System Firmware	N	1	○	○	2	○	-		AttkFinder [39]	Process Vul. Discovery	E	1	1	1	●	96	PR	●	-
	Compromise System Functions [221]		SM	1	○	○	3	○	-		VETPLC [216]		E	1	1	1	○	N/S	PR	○	O
	Decrypting Siemens Simatic firmware [31]		P	1	○	●	2	◑	-		Similo [155]		E	1	2	2	○	100	RE	●	O
	Firmware leakage [35]		U	3	○	○	1	○	-		Logging input simulation [200]		E	1	2	1	○	99.9	DE	○	-
	Firmware modification attack [27]		U	3	○	○	3	○	-		CPAC [63]	Auth. Enforcement	X	2	2	2	○	N/S	PR	○	-
	Persistent denial-of-service attack [167]	Modify Program	E	0	○	○	2	○	-		Firmware verification tool [134]	Code Signing	X	1	1	1	○	N/S	PR	○	-
	Remotely-triggered DoS [167]		E	0	○	○	2	○	-												
	Time-based denial-of-service attack [167]		E	0	○	○	2	○	-		PLCDefender [160]	Attestation	E	2	3	1	○	98	PR	●	O
	Flash Update [161]	Activate Firmware Update Mode	E	0	○	●	2	○	-												
	HARVEY [75]	Data from Debug Port	MC	3	●	○	3	○	-												
	Shutting down the PLC [82]	Device Restart/Shutdown	N	1	○	○	2	○	-												
	Rogue Firmware Load [150]	Module Firmware	E	1	○	○	2	●	-		SNIFU [30]	Code Signing	U	2	2	1	○	100	PR	○	-
I/O	Leak I/O Process Data [193]	Exfiltration over Side-channel	N	2	○	○	1	-	-		ORRIS [156]	Antivirus/Antimalware	X	2	2	1	●	85.6	DE	○	C
	Blinkware [192]		X	3	◑	○	1	○	-												
	Analog Emissions attack [114]		N	3	◑	○	1	○	-												
	PHYCO [76]		U	3	◑	○	1	○	-												
	Escalated Privilege I/O Command [87]	Adversary-in-the-Middle	E	2	○	○	2	○	-		Hidden Sensor Measurements [81]	Network Intrusion Detection	X	1	2	1	○	N/S	DE	○	-
	I/O Command attack [87]		E	1	○	○	2	○	-		Physics-based attack detection [196]		E	1	1	1	○	N/S	DE	○	-
	Wireless Control [68]	Spoof Reporting Message	N	3	○	○	2	◑	-		Blockchain Monitoring [42]	Encrypt Network Traffic	E	1	3	2	○	N/S	PR	○	-
	Malicious Expanders [68]		N	3	○	○	2	○	-												
	False sequence attack [207]	Brute Force I/O	N	2	●	○	3	-	-		PCAT [172]	Validate Program Inputs	E	1	2	1	○	N/S	PR	-	-
	Manipulated Variable [12]	Manipulation of Control	X	2	◑	○	2	○	-		Smart I/O Modules [149]		M	1	2	1	●	N/S	PR	-	-
	OPC UA Supply Chain Attack [94]	Supply Chain Compromise	UA	0	◑	○	2	○	-		PAtt [79]	Attestation	X	2	2	1	○	97	PR	○	-
	Backdoor Attack [199]	Unauthorized Command Message	N	2	○	●	2	-	-												
Memory	Leak Crypto secrets [193]	Exfiltration over Side-channel	N	1	○	○	1	-	-												
	WaterLeakage [157]		S7	0	○	○	1	○	-												
	PLC-Blaster [180]	Change Operating Mode	S7	0	○	○	2	○	-		Mitigate Malicious Disruption [40]	Redundancy of Service	N	2	1	1	○	N/S	DE	-	O
	Clear PLC memory [202]	Data Destruction	P	0	○	○	2	○	-		CPS Twinning [59]		M	1	3	3	●	N/S	RE	○	-
	ICS-BROCK [218]	Data Encrypted for Impact	U	3	○	○	2	○	-		Armor PLC [217]		N	1	3	1	○	N/S	DE	-	O
	Memory Dump [221]	Data from Debug Port	SM	1	○	○	1	○	-												
	Exfiltrate FB Variables [88]	Exfiltration over ICS Protocol	P	0	○	○	1	◑	-												
	Storage Based Covert Channel [88]	Fallback Channels	P	0	○	○	1	◑	-												
	DB content manipulation [126]	Modify Parameter	S7	0	◑	○	2	○	-												
	ROP Attack [26]	Process Injection	M	0	○	○	2	○	-												
	False Command Injection Attack [14]	Unauthorized Command Message	M	0	○	○	2	○	O												
CPU	Forcing a CPU Stop [161]	Device Restart/Shutdown	E	0	○	●	2	○	-		WeaselBoard [141]	Exploit Protection	BP	3	3	3	○	N/S	PR	●	-
	Crash CPU [161]		E	0	○	●	2	○	-		C2 [130]		N	2	2	1	○	N/S	PR	○	-
	CPU Stop and Start Attack [31]		P	0	○	●	2	◑	-												
	S7-1200 Start/Stop Attack [34]	Change Operating Mode	S7	0	○	○	2	◑	-												
	S7-1500 Start/Stop Attack [34]		S7	0	○	○	2	○	-												
	ASIC Reverse Engineering	Supply Chain Compromise	X	3	○	○	3	○	-												

Table 2: A Summary of PLC Attack and Defense Methods (Control Logic, Firmware, I/O, Memory and CPU Components).
S7=S7COMM, **UA**=OPC-UA, **P**=Profinet, **E**=EtherNet/IP, **M**=Modbus TCP, **T**=TriStation, **SM**=SoMachine, **U**=USB Port, **MC**=Memory Card, **X**=Others, **O**=OpenPLC, **C**=CODESYS, **N**=Not Specified, **DE**=Detection, **PR**=Prevention, **RE**=Recovery

2) how to hide the infection from the engineering workstation. To address challenge 1, *Yoo and Ahmed* [213] propose two control logic infection attacks that can bypass network intrusion detection systems. In the first attack, they bypass intrusion detection systems that look for transfers of control logic (compile code) by injecting control code in data blocks (used for the transfer of data such as counters) and then modifying the execution pointer to execute code in data blocks. The second attack uses fragmentation and noise to further obfuscate these control logic transfers to the PLC.

Kalle et al. [105] target both challenges (infection and stealthiness). They consider a different problem in code injection, where they assume they have a binary they want to modify. They then develop a decompiler transforming low-level control logic to a high-level instruction list to help them inject the malicious code before recompiling it into a binary that can be uploaded to the vulnerable PLC. They also develop a virtual PLC that interfaces with the engineering workstation (via an AiTM attack); this virtual PLC then sends previously captured network traffic of the original uninfected control logic back to the workstation.

Attacks that Target the PLC CPU. As discussed in Appendix E, PLCs generally have three CPU operating modes: "STOP," "RUN," and "PROGRAM." Attacks targeting the PLC CPU focus on disabling the CPU remotely (AL3 or AL2) with STOP commands. One of the earliest examples of these attacks launched against Siemens PLCs by impersonating the workstation is the work of Beresford [31]. While Siemens released cryptographic protections for these connections that would prevent these attacks, more recent work reverse-engineered the cryptographic protocol. *Biham et al.* [34] were able to create a rogue engineering station that could remotely start or stop these newer PLCs by compromising a vulnerable key exchange protocol.

Passwords can protect CPU modes of operation, but a compromised password can still enable remote attacks. Higher-end PLCs protect their CPU modes with physical methods, such as using a physical switch or a physical key. These methods are further discussed in Appendix E.

Attacks that Target the PLC Firmware. Firmware modification is one of the most powerful attacks in any platform, as the attacker can control access to the input and output modules of the PLC while remaining undetected. *Basnight et al.* [27] pioneered methods on PLC firmware reverse-engineering and how to develop modified firmware as a proof of concept.

The most cited firmware attack paper in our study is Harvey [75]. The authors extracted firmware images from the update packages of the PLC vendor's website and the PLC memory through the JTAG interface of the PLC processor. Then they identified the subroutines that allowed them to modify the inputs and outputs for the PLC. To upload the firmware, they rely on vulnerabilities to protect remote firmware update functions (AL0, AL1 or AL2 in Fig. 4) or directly through the JTAG interface (AL3 in Fig. 4).

Attacks that Target the PLC I/O. The two most cited papers targeting the inputs and outputs of the PLC focus on how to use input or output physical signals as covert channels to exchange information. For example, *PHYCO* [76] proposes a method where two compromised PLCs can talk to each other, even when a firewall exists between them. One PLC can send an output to increase power generation, and the other PLC can read the increased generation as a bit of information. Adversaries can also use the PLC I/O to exfiltrate data. *Krishnamurthy et al.* [114] illustrate how a PLC sending control commands to a motor can leak information about the status of a chemical plant.

Attacks that Target the PLC Runtime. An illustrative example of runtime attacks is provided by *Abbasi et al.* [2], by introducing Pin Control Attacks. This attack involves tampering with the SoC configuration within the PLCs, aiming to disrupt the communication between the PLC runtime and the hardware peripherals. By implementing this attack, the adversary severs the connection between the runtime software and the physical world, allowing them to manipulate the control data within the actual physical process.

Runtime attacks can also target the availability of the system; for example, *Gjendemsjø et al.* [82] proposed an XML Bomb Attack that causes the PLC runtime to crash by modifying an XML file.

Attacks that Target the PLC Operating System. There are not a lot of papers focusing on attacking the OS of PLCs. In their comprehensive study of the attack surface of a PLC, *Abbasi et al.* [4] touch upon an intriguing aspect of PLC security: the vulnerability of the Siemens Adonis Real-Time Operating System. Although their primary focus lies on the bootloader security of Siemens PLCs, they also explore security weaknesses inherent in the Adonis RTOS.

Attacks that Target the PLC Memory. Some network or control logic attacks target the memory of the PLC as part of their infection chain by modifying memory blocks or by getting memory dumps. The most cited efforts focused on developing worms stored in memory [157, 180].

So far, we have introduced the most cited examples of attacks per target, but we have not illustrated representative defenses. Because of limited space, we refer the reader to a discussion of defenses in Appendix G.

6 Research Gaps

We now focus on identifying relevant insights, gaps, and recommendations for future work by analyzing the data from Tables 1 and 2.

Most of the Attacks Require Zero Environment Knowledge. We found that 82% (97/119) of the attacks that may cause either *limited* (1) (36% (35/97)), *substantial* (2) (56% (54/97)), or *severe* (3) (8% (8/97)) damage require *zero knowledge* (O) of the environment in which the PLC lives,

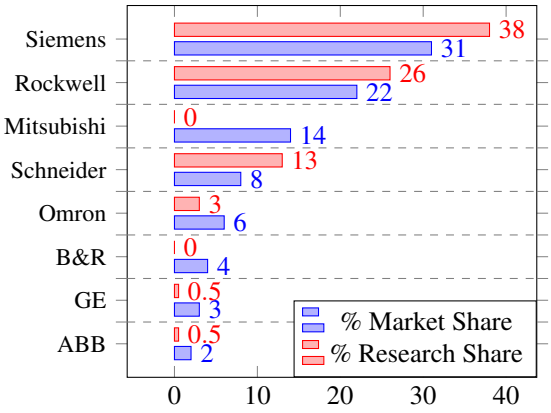

Figure 5: Comparison of Research Share vs Market Share.

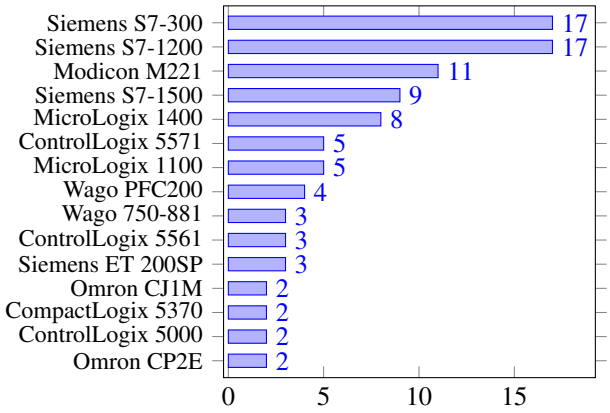

Figure 6: The 15 Most Common PLC Models.

following the description presented in Sec. 2.1 and the criteria discussed in Sec. 4.3. This may imply that adversaries can potentially launch non-trivial attacks against power grids, chemical plants, water treatment plants, etc. by targeting PLCs without having to invest time performing reconnaissance or learning the specifics of actuators, sensors, etc. Moreover, we were able to identify that only a few defense methods, e.g., *Formby et al.* [69] and *Bellettini et al.* [28], make use of such important environmental information. We therefore recommend that future defenses make use of environment knowledge in their strategy to increase their effectiveness.

The Security of Important PLC Brands Has Not Been Explored. We found that the security of some important PLC platforms widely used in practice has been ignored. As shown in Fig. 5, the *market* share percentage of important PLC manufacturers such as Mitsubishi, Omron, ABB, and GE is considerably higher than their *research* share, e.g., the number of papers explicitly using them for attack/defense evaluation purposes. For example, even though Mitsubishi PLCs account for 14% of the global market, they contribute to 0% of the research share in our review (they only appear once in our Grey literature survey [41]). Also, Omron PLCs account for

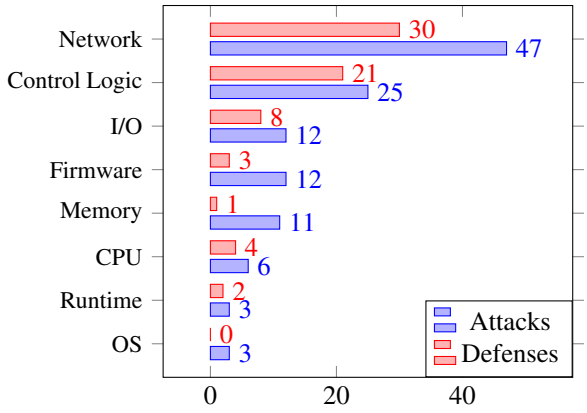

Figure 7: Defense vs Attack Methods per Target Component.

6% of the market share, yet they appear in only 3% of our research, and ABB PLCs account for 2% of the global market share, yet they appear in only 0.5% of the research share. Finally, GE PLCs hold 3% of the market share, yet they appear in only 0.5% of the literature. This means that there are potentially thousands of PLCs deployed in the world whose security has not been explored or may not have been adequately understood. We therefore recommend future lines of work specifically addressing these devices, exploring the effectiveness of existing attacks and defenses as discussed in this work, as well as the introduction of newer security methods tailored for them.

Lack of Defenses at the Recovery Stage. We found that most defense methods are designed for the Prevention (PR) and Detection (DE) stages discussed in Sec. 4.7, 47% (33/70) and 47% (33/70) out of 70 respectively, whereas the Recovery (RE) stage accounts for only 6% (4/70). This means that in the event of a successful attack there are limited options to recover and bring the PLC back to operation.

Most Attacks and Defenses are Evaluated on a Small Subset of PLCs. Our results show that the top 5 most common PLC models in our literature review (as shown in Fig. 6) account for 40% of all studies. This may result in a narrow understanding of PLC security that excludes the rest of the PLC models in the market. Additionally, we found that 80% (95/119) of attack methods and 81% (57/70) of defense methods were evaluated using a single HardPLC model (O). This means that most attack and defense methods are shown to work with a single HardPLC model, making it unclear whether or not the defense method can be generalized to other PLC models and manufacturers. Therefore, we recommend that future research include an evaluation of multiple PLCs.

Important Tactics have Little to No Research. Our ICS^2 Matrix includes 14 Tactics. However, as Fig. 8 shows, most of the attack methods focus on Impair Process Control, Inhibit Response Function, Exfiltration, Collection, and Persistence. On the other hand, important Tactics like Command and Con-

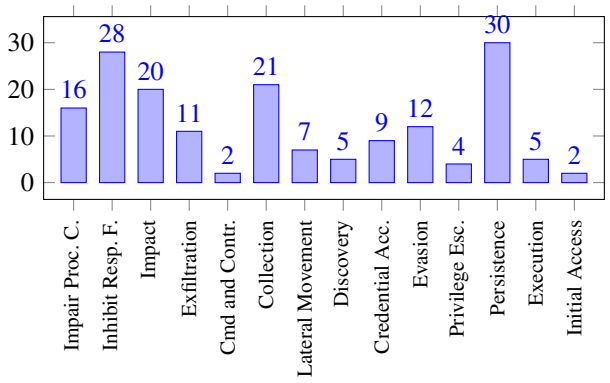

Figure 8: Attack Methods according to their Tactic.

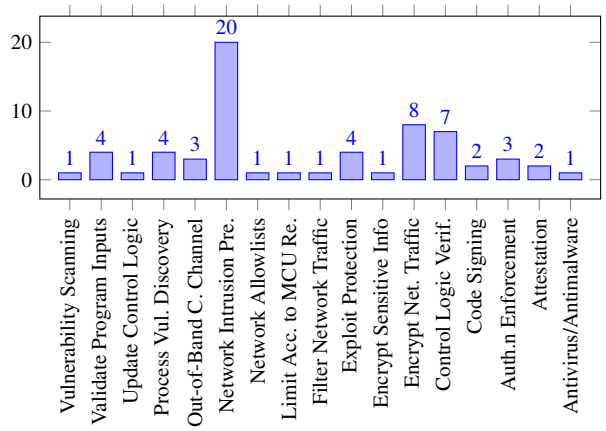

Figure 9: Defense Methods per Mitigation Category.

trol, Lateral Movement, Evasion, Credential Access, and Initial Access have not been investigated. One reason that might explain why these tactics have not been explored in previous research is that they are too specific or custom-fit. For example, the Command and Control tactic includes techniques such as *Commonly Used Port* and *Standard Application Layer Protocol*, which depend on specific network protocols and ports. This specific focus on network ports might limit the scope of research that can be carried out for this tactic.

Most Mitigation Strategies have Little to No Research. We found defense methods that match 17 Mitigation categories. However, as shown in Fig.9, the majority of these defenses fall into the Network Intrusion Prevention, Encrypt Network Traffic and Control Logic Verification categories. The remaining 14 Mitigation categories have only 4 or fewer defenses. These include important categories like Antivirus/Antimalware, Vulnerability Scanning, and Attestation, which are especially important given the rise of ICS malware such as the ones described in Appendix F.

Weaknesses of State-of-the-Art Defenses Based on the results shown in Tables 1 and 2, we identified the following three major weaknesses when it comes to defending PLCs: **1)** No ransomware detection. Even though researchers have shown that ransomware attacks against PLCs are possible [70, 218] and there are multiple documented ransomware attacks against ICS, there is no available research focused on how to detect and stop PLC ransomware. To address this research gap, future research should introduce new defense methods that take advantage of state-of-the-art malware and ransomware detection techniques such as sandbox detection [197] and static and dynamic analysis [148]. **2)** No web-based malware detection. Research shows that attackers target PLCs' web interfaces [122, 164] and that it is possible to compromise PLCs via their web applications [151]. However, there is no available research focused on how to detect and stop malware targeting PLCs' web interfaces. **3)** No Exfiltration over Covert Channel Detection. This is one of the new techniques that we introduced in our ICS^2 Matrix (Fig. 13), which includes methods such as *PHYCO* [76]. Introducing this technique lays the groundwork for identifying the need for mitigations against such techniques. Currently, there is no known mitigation for this type of attack. Indeed, in 2014 *Garcia et al.* wrote: "There has been no detection solution capable of identifying such hidden communications in the physical power system," which still holds true.

7 Discussion

In this section, we discuss research challenges that haven't attracted enough attention and may become relevant as PLCs evolve.

Reproducible Research. Based on our analysis and the results presented in Tables 1 and 2, few defense and attack methods provide publicly available research artifacts. During our literature review, we searched for research artifacts for each paper. We searched the paper itself on Google and the author's website. Using this method, we were able to find the source code for 19 papers (only 16%). This limits the reproducibility of attacks and defenses. In an attempt to find the artifacts for papers without openly available code, we contacted 91 authors via email requesting their research artifacts, and we received 16 responses (17.6%). Ultimately only 3 shared a research artifact. The other 13 did not share the source code for the following reasons: **1)** The project was completed long ago or the first author moved on to a different institution (30%). **2)** There were funding or distribution restrictions (25%). **3)** The authors were working on it and will publish it later (15%). **4)** There were no plans to release it to the public (30%). 16% is a low number of papers with artifacts, and this does not even consider if the source code has good documentation or if these public resources are easy to run. Therefore, we encourage researchers to release their PLC security artifacts so that research can be replicated and built upon. We encourage researchers to leverage our PLC security artifacts repository discussed in Sec. 1 to disseminate their artifacts. We acknowledge that given the criticality of

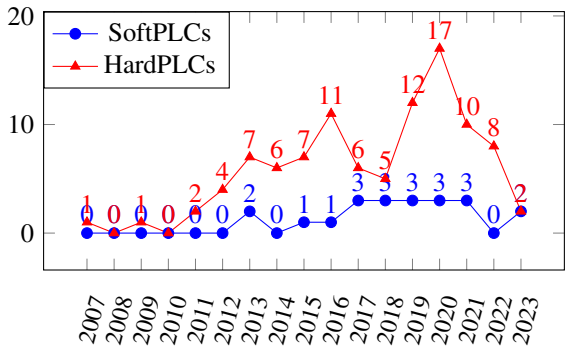

Figure 10: Papers evaluated using SoftPLCs vs HardPLCs.

PLCs it is not always possible to release artifacts at the time of publication if at all. For example, the US' Cybersecurity & Infrastructure Security Agency (CISA) recommends researchers "provide us a reasonable amount of time to resolve the issue before you disclose it publicly [198]."

A second challenge that compounds the problem of lack of reproducible research is the absence of standardized performance metrics. Based on the results in Tables 1 and 2, few defense methods reported detailed PLC overhead or effectiveness metrics as described in Sec. 4.7. This makes comparing and building upon previous research almost impossible. To overcome this challenge we recommend that future defense methods report detailed quantifiable metrics. First, papers that propose methods such as Intrusion Detection Systems (IDS) should report a complete and detailed confusion matrix. The work by *Salehi et al.* [160] provides an excellent example. Second, future research on defense methods should report detailed and quantifiable benchmark metrics, e.g., memory overhead. Existing benchmarks such as *BenchmarkIoT* [13] can be customized for PLCs to overcome this challenge.

Transition from HardPLCs to SoftPLCs. As discussed in the introduction, PLCs are going through a paradigm shift with support for new protocols and new functionalities. Support for SoftPLCs appears to be increasing. While SoftPLCs currently have a minimal market share, large manufacturers like Siemens and Rockwell are preparing hardware-agnostic products [178]. This paradigm shift has also reached the scientific community. As shown in Fig. 10, PLC security research has diverged into two different strands: evaluations that use HardPLCs and evaluations that use SoftPLCs.

Considering this trend, future research should be aimed at addressing the following challenges: **1)** Developing transitional defense methods that secure both HardPLCs and SoftPLCs. For example, developing *bump-in-the-wire* solutions that are compatible with HardPLCs and SoftPLCs. This may involve evaluating research on both HardPLCs and SoftPLCs. **2)** Investigating defense mechanisms available for SoftPLCs that were previously unavailable for HardPLCs. For example, HardPLCs' proprietary nature limited their access control and encryption capabilities by incorporating weak encryption protocols. SoftPLCs' less opaque architecture provides the opportunity to solve this problem. **3)** Investigating both attack and defense methods that are possible only with SoftPLCs. For example, CODESYS' SoftPLCs can be connected to the cloud [45], which opens new ways to collect data that can be used to train machine learning-based Intrusion Detection Systems.

8 Related Work

SoKs on PLCs. To the best of our knowledge, the work by *Sun et al.* [185] is the only Systematization of Knowledge focused on attacks and defenses on the control logic of PLCs. However, our work is different in two ways. First, *Sun et al.* focuses on the Control Logic of PLCs, while our work takes a more comprehensive approach that includes 8 additional PLC components. Second, our SoK includes papers with practical evaluations that include experiments with PLCs, while *Sun et al.* focus on formal or theoretical research.

PLC Honeypots. While previous work includes approaches for PLC honeypots [72, 122], their focus is intelligence gathering rather than attacking PLCs.

Fuzzing and Binary Reverse Engineering. There is also noticeable research on fuzzing and reverse engineering for ICS and PLCs in the literature [66, 107, 145, 191, 201]. Our work does not cover this topic, because our focus is not software testing but outlining specific attacks.

9 Conclusion and Future Work

In this paper, we provide a systematic review of currently-known attacks and defenses for PLCs. We also pointed out research gaps that should be tackled in the future so that the security of PLCs can be better understood, thus helping avoid future attacks against ICS and PLCs. We hope this systematization is useful to newcomers to the field as well as experienced PLC researchers looking to contextualize their work. As a part of Future Work, we are developing an experimental testbed as discussed in Sec. 2.1, which is in turn composed of different PLC models and brands. This way, many different attack and defense strategies can be launched to provide experimental evidence on their effectiveness, deployability, and robustness, thus ultimately complementing the results provided as a part of this work.

References

[1] Abbasi, A., and Genuise, A. Ghost in the plc vs ghostbuster: on the feasibility of detecting pin control attack in programmable logic controllers. In *Ghost in the PLC vs GhostBuster*. Eindhoven University of Technology, 2017.

[2] Abbasi, A., and Hashemi, M. Ghost in the plc designing an undetectable programmable logic controller rootkit via pin control attack. *Black Hat Europe 2016* (2016), 1–35.

[3] ABBASI, A., HOLZ, T., ZAMBON, E., AND ETALLE, S. Ecfi: Asynchronous control flow integrity for programmable logic controllers. In *Proceedings of the 33rd Annual Computer Security Applications Conference* (2017), pp. 437–448.

[4] ABBASI, A., SCHARNOWSKI, T., AND HOLZ, T. Doors of Durin: The Veiled Gate to Siemens S7 Silicon. *Black Hat Europe* (2019), 53.

[5] ADEPU, S., SHRIVASTAVA, S., AND MATHUR, A. Argus: An orthogonal defense framework to protect public infrastructure against cyber-physical attacks. *IEEE Internet Computing 20*, 5 (2016), 38–45.

[6] AG, S. System software for s7-300/400 system and standard functions volume 1/2. https://cache.industry.siemens.com/dl/files/604/44240604/att_67003/v1/s7sfc_en-EN.pdf. Accessed: 27-09-2023.

[7] AG, S. Training document for the company-wide automation solution totally integrated automation (tia). https://www.automation.siemens.com/sce-static/learning-training-documents/classic/advanced-programming/b04-data-blocks-en.pdf. Accessed: 27-09-2023.

[8] AHMED, C. M., OCHOA, M., ZHOU, J., AND MATHUR, A. Scanning the cycle: Timing-based authentication on plcs. In *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security* (New York, NY, USA, 2021), ASIA CCS '21, Association for Computing Machinery, p. 886–900.

[9] AL FAROOQ, A., MARQUARD, J., GEORGE, K., AND MOYER, T. Detecting safety and security faults in plc systems with data provenance. In *2019 IEEE International Symposium on Technologies for Homeland Security (HST)* (2019), IEEE, pp. 1–6.

[10] ALESSANDRO ERBA, ANNE MÜLLER, N. O. T. Resting on feet of clay: Securely bootstrapping opc ua deployments. *Black Hat Europe* (2021). https://i.blackhat.com/EU-21/Wednesday/EU-21-Erba-Resting-On-Feet-Of-Clay-Securely-Bootstrapping-OPC-UA-Deployments.pdf. Accessed: 27-09-2023.

[11] ALEXANDER, O. In Pursuit of a Gestalt Visualization: Merging MITRE ATT&CK® for Enterprise and ICS to Communicate. *MITRE ATT&CK®* (Sept. 2020). https://medium.com/mitre-attack/in-pursuit-of-a-gestalt-visualization-merging-mitre-att-ck-for-enterprise-and-ics-to-communicate-3523daa7b580. Accessed: 27-09-2023.

[12] ALEXANDER BOLSHEV, M. K. Never trust your inputs: Causing 'catastrophic physical consequences' from the sensor (or how to fool adc). *Black Hat Asia* (2016). https://www.blackhat.com/docs/asia-16/materials/asia-16-Bolshev-Never-Trust-Your-Inputs-Causing-Catastrophic-Physical-Consequences-From-The-Sensor.pdf. Accessed: 27-09-2023.

[13] ALMAKHDHUB, N. S., CLEMENTS, A. A., PAYER, M., AND BAGCHI, S. Benchiot: A security benchmark for the internet of things. In *2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)* (2019), pp. 234–246.

[14] ALSABBAGH, W., AMOGBONJAYE, S., URREGO, D., AND LANGENDÖRFER, P. A stealthy false command injection attack on modbus based scada systems. In *2023 IEEE 20th Consumer Communications & Networking Conference (CCNC)* (2023), IEEE, pp. 1–9.

[15] ALSABBAGH, W., KIM, C., AND LANGENDÖRFER, P. Good night, and good luck: A control logic injection attack on openplc. Tech. rep., EasyChair, 2023.

[16] ALSABBAGH, W., AND LANGENDÖRFER, P. Patch now and attack later-exploiting s7 plcs by time-of-day block. In *2021 4th IEEE International Conference on Industrial Cyber-Physical Systems (ICPS)* (2021), IEEE, pp. 144–151.

[17] ALSABBAGH, W., AND LANGENDÖRFER, P. A stealth program injection attack against s7-300 plcs. In *2021 22nd IEEE International Conference on Industrial Technology (ICIT)* (2021), vol. 1, IEEE, pp. 986–993.

[18] ALSABBAGH, W., AND LANGENDÖRFER, P. No need to be online to attack-exploiting s7-1500 plcs by time-of-day block. In *2022 XXVIII International Conference on Information, Communication and Automation Technologies (ICAT)* (2022), IEEE, pp. 1–8.

[19] ALVES, T. OpenPLC Runtime version 3. https://github.com/thiagoralves/OpenPLC_v3, Sept. 2023. Accessed: 27-09-2023.

[20] ALVES, T., DAS, R., AND MORRIS, T. Embedding encryption and machine learning intrusion prevention systems on programmable logic controllers. *IEEE Embedded Systems Letters 10*, 3 (2018), 99–102.

[21] ALVES, T., MORRIS, T., AND YOO, S.-M. Securing scada applications using openplc with end-to-end encryption. In *Proceedings of the 3rd annual industrial control system security workshop* (2017), IEEE, pp. 1–6.

[22] ALVES, T. R., BURATTO, M., DE SOUZA, F. M., AND RODRIGUES, T. V. Openplc: An open source alternative to automation. In *IEEE Global Humanitarian Technology Conference (GHTC 2014)* (2014), IEEE, pp. 585–589.

[23] AUTOMATION, R. Logix 5000 Controllers Security. https://literature.rockwellautomation.com/idc/groups/literature/documents/pm/1756-pm016_-en-p.pdf, 2022. Accessed: 27-09-2023.

[24] AUTOMATION, R. Logix 5000 Controllers Tasks, Programs, and Routines. https://literature.rockwellautomation.com/idc/groups/literature/documents/pm/1756-pm005_-en-p.pdf, 2022. Accessed: 27-09-2023.

[25] AYUB, A., YOO, H., AND AHMED, I. Empirical study of plc authentication protocols in industrial control systems. In *2021 IEEE Security and Privacy Workshops (SPW)* (2021), IEEE, pp. 383–397.

[26] AYUB, A., ZUBAIR, N., YOO, H., JO, W., AND AHMED, I. Gadgets of gadgets in industrial control systems: Return oriented programming attacks on plcs. In *2023 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)* (2023), IEEE, pp. 215–226.

[27] BASNIGHT, Z., BUTTS, J., LOPEZ JR, J., AND DUBE, T. Firmware modification attacks on programmable logic controllers. *International Journal of Critical Infrastructure Protection 6*, 2 (2013), 76–84.

[28] BELLETTINI, C., AND RRUSHI, J. Scada protocol obfuscation: A proactive defense line in scada systems. In *SCADA Security Scientific Symposium* (2007), vol. 154, Digital Bond Press.

[29] BELLETTINI, C., AND RRUSHI, J. L. Vulnerability analysis of scada protocol binaries through detection of memory access taintedness. In *2007 IEEE SMC Information Assurance and Security Workshop* (2007), IEEE, pp. 341–348.

[30] BENKRAOUDA, H., CHAKKANTAKATH, M. A., KELIRIS, A., AND MANIATAKOS, M. Snifu: Secure network interception for firmware updates in legacy plcs. In *2020 IEEE 38th VLSI Test Symposium (VTS)* (2020), IEEE, pp. 1–6.

[31] BERESFORD, D. Exploiting siemens simatic s7 plcs. *Black Hat USA 16*, 2 (2011), 723–733. https://media.blackhat.com/bh-us-11/Beresford/BH_US11_Beresford_S7_PLCs_WP.pdf. Accessed: 27-09-2023.

[32] BHATIA, S., KUSH, N. S., DJAMALUDIN, C., AKANDE, A. J., AND FOO, E. Practical modbus flooding attack and detection. In *Proceedings of the Twelfth Australasian Information Security Conference (AISC 2014)[Conferences in Research and Practice in Information Technology, Volume 149]* (2014), Australian Computer Society, pp. 57–65.

[33] BIALLAS, S., BRAUER, J., AND KOWALEWSKI, S. Arcade. plc: A verification platform for programmable logic controllers. In *2012 Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering* (2012), IEEE, pp. 338–341.

[34] BIHAM, E., BITAN, S., CARMEL, A., DANKNER, A., MALIN, U., AND WOOL, A. Rogue7: Rogue engineering-station attacks on s7 simatic plcs. *Black Hat USA* (2019). https://i.blackhat.com/USA-19/Thursday/us-19-Bitan-Rogue7-Rogue-Engineering-Station-Attacks-On-S7-Simatic-PLCs-wp.pdf. Accessed: 27-09-2023.

[35] BITAN, S., AND DANKNER, A. sOfT7: Revealing the Secrets of the Siemens S7 PLCs. *Black Hat USA* (2022), 36. https://i.blackhat.com/USA-22/Wednesday/US-22-Bitan-Revealing-S7-PLCs.pdf. Accessed: 27-09-2023.

[36] BOLTON, W. *Programmable logic controllers*. Newnes, 2015.

[37] BONNEY, G., HÖFKEN, H., PAFFEN, B., AND SCHUBA, M. Ics/scada security analysis of a beckhoff cx5020 plc. In *2015 International Conference on Information Systems Security and Privacy (ICISSP)* (2015), IEEE, pp. 1–6.

[38] BRUSSO, B. C. 50 years of industrial automation [history]. *IEEE Industry Applications Magazine 24*, 4 (2018), 8–11.

[39] CASTELLANOS, J. H., OCHOA, M., CARDENAS, A. A., ARDEN, O., AND ZHOU, J. Attkfinder: Discovering attack vectors in plc programs using information flow analysis. In *24th International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2021)* (San Sebastian, Spain, 2021), RAID '21, Association for Computing Machinery, p. 235–250.

[40] CHATTERJEE, U., SANTIKELLUR, P., SADHUKHAN, R., GOVINDAN, V., MUKHOPADHYAY, D., AND CHAKRABORTY, R. S. United we stand: A threshold signature scheme for identifying outliers in plcs. In *2019 56th ACM/IEEE Design Automation Conference (DAC)* (2019), IEEE, pp. 1–2.

[41] CHENG, M., AND YANG, S. Taking Apart and Taking Over ICS & SCADA Ecosystems: A Case Study of Mitsubishi Electric. *DEF CON 29* (2021). https://media.defcon.org/DEF%20CON%2029/DEF%20CON%2029%20presentations/Mars%20Cheng%20Selmon%20Yang%20-%20Taking%20Apart%20and%20Taking%20Over%20ICS%20%26%20SCADA%20Ecosystems%20-%20%20A%20Case%20Study%20of%20Mitsubishi%20Electric.pdf. Accessed: 03-10-2023.

[42] CHOI, M. K., YEUN, C. Y., AND SEONG, P. H. A novel monitoring system for the data integrity of reactor protection system using blockchain technology. *IEEE Access 8* (2020), 118732–118740.

[43] CHOI, T., BAI, G., KO, R. K., DONG, N., ZHANG, W., AND WANG, S. An analytics framework for heuristic inference attacks against industrial control systems. In *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)* (2020), IEEE, pp. 827–835.

[44] Cloud Connectivity – IoT PLC: Controllers with MQTT. https://www.wago.com/us/open-automation/cloud-connectivity/iot-plc-controller-with-mqtt. Accessed: 27-09-2023.

[45] Codesys automation server | codesys store international. https://store.codesys.com/en/codesys-automation-server.html. Accessed: 08-11-2023.

[46] CODESYS Inside. https://www.codesys.com/the-system/codesys-inside.html. Accessed: 27-09-2023.

[47] CODESYS Runtime. https://www.codesys.com/products/codesys-runtime.html. Accessed: 27-09-2023.

[48] CONN, V. S., VALENTINE, J. C., COOPER, H. M., AND RANTZ, M. J. Grey literature in meta-analyses. *Nursing research 52*, 4 (2003), 256–261.

[49] ControlLogix System User Manual. https://literature.rockwellautomation.com/idc/groups/literature/documents/um/1756-um001_-en-p.pdf. Accessed: 02-10-2023.

[50] COOK, M. M., MARNERIDES, A. K., AND PEZAROS, D. Plcprint: Fingerprinting memory attacks in programmable logic controllers. *IEEE Transactions on Information Forensics and Security 18* (2023), 3376–3387.

[51] Cyber-Physical Systems Workshop. https://web.archive.org/web/20080517071555/http:/varma.ece.cmu.edu/cps/, 2006. Accessed: 27-09-2023.

[52] CYBERSECURITY, AND (CISA), I. S. A. Apt cyber tools targeting ics/scada devices. https://www.cisa.gov/uscert/ncas/alerts/aa22-103a, 2022. Accessed: 02-10-2023.

[53] DAVID, A., AND GEORGE, L. Exfiltrating reconnaissance data from air-gapped ics/scada netowrks. *Black Hat Europe* (2017). https://www.blackhat.com/docs/eu-17/materials/eu-17-Atch-Exfiltrating-Reconnaissance-Data-From-Air-Gapped-Ics-Scada-Networks.pdf. Accessed: 02-10-2023.

[54] DIETRICH, D., NEUMANN, P., AND SCHWEINZER, H. F. Fieldbus technology: Systems integration, networking, and engineering. In *Proceedings of the Fieldbus Conference FeT'99 in Magdeburg, Federal Republic of Germany* (1999), Springer.

[55] DIETZ, M., VIELBERTH, M., AND PERNUL, G. Integrating digital twin security simulations in the security operations center. In *Proceedings of the 15th International Conference on Availability, Reliability and Security* (New York, NY, USA, 2020), ARES '20, Association for Computing Machinery.

[56] DRAGOS, A. Trisis malware: Analysis of safety system targeted malware. https://www.dragos.com/resource/trisis-analyzing-safety-system-targeting-malware/, 2017. Accessed: 27-09-2023.

[57] DRAGOS, I. Chernovite's pipedream malware targeting industrial control systems (ics). https://www.dragos.com/blog/industry-news/chernovite-pipedream-malware-targeting-industrial-control-systems/, 2022. Accessed: 27-09-2023.

[58] DUNLAP, S., BUTTS, J., LOPEZ, J., RICE, M., AND MULLINS, B. Using timing-based side channels for anomaly detection in industrial control systems. *International Journal of Critical Infrastructure Protection 15* (2016), 12–26.

[59] ECKHART, M., AND EKELHART, A. Towards security-aware virtual environments for digital twins. In *Proceedings of the 4th ACM Workshop on Cyber-Physical System Security* (New York, NY, USA, 2018), CPSS '18, Association for Computing Machinery, p. 61–72.

[60] Encrypted block protection for fbs and fcs from step 7 v5.5. https://support.industry.siemens.com/cs/document/45632073/how-do-you-install-the-improved-block-protection-for-fbs-and-fcs-in-step-7-v5-5-onwards-?dti=0&lc=en-AO. Accessed: 02-10-2023.

[61] ERBA, A., TAORMINA, R., GALELLI, S., POGLIANI, M., CARMINATI, M., ZANERO, S., AND TIPPENHAUER, N. O. Constrained concealment attacks against reconstruction-based anomaly detectors in industrial control systems. In *Annual Computer Security Applications Conference* (New York, NY, USA, 2020), ACSAC '20, Association for Computing Machinery, p. 480–495.

[62] EtherNet/IP™ | ODVA Technologies | Industrial Automation. https://www.odva.org/technology-standards/key-technologies/ethernet-ip/. Accessed: 27-09-2023.

[63] ETIGOWNI, S., TIAN, D. J., HERNANDEZ, G., ZONOUZ, S., AND BUTLER, K. Cpac: Securing critical infrastructure with cyber-physical access control. In *Proceedings of the 32nd Annual Conference on Computer Security Applications* (New York, NY, USA, 2016), ACSAC '16, Association for Computing Machinery, p. 139–152.

[64] Examples for the SIMATIC S7-1200 / S7-1500 Web Server. https://cache.industry.siemens.com/dl/files/496/68011496/att_959527/v2/68011496_Examples_for_S7WebServer_DOC_v21_en.pdf, 2018. Accessed: 02-10-2023.

[65] FALLIERE, N., MURCHU, L. O., AND CHIEN, E. W32. stuxnet dossier. *White paper, Symantec Corp., Security Response 5*, 6 (2011), 29.

[66] FANG, D., SONG, Z., GUAN, L., LIU, P., PENG, A., CHENG, K., ZHENG, Y., LIU, P., ZHU, H., AND SUN, L. Ics3fuzzer: A framework for discovering protocol implementation bugs in ics supervisory software by fuzzing. In *Annual Computer Security Applications Conference* (New York, NY, USA, 2021), ACSAC '21, Association for Computing Machinery, p. 849–860.

[67] FELSER, M., AND SAUTER, T. The fieldbus war: history or short break between battles? In *4th IEEE international workshop on factory communication systems* (2002), IEEE, pp. 73–80.

[68] FITZPATRICK, J. The tao of hardware, the te of implants. *Black Hat, USA* (2016). https://www.blackhat.com/docs/us-16/materials/us-16-FitzPatrick-The-Tao-Of-Hardware-The-Te-Of-Implants-wp.pdf. Accessed: 02-10-2023.

[69] FORMBY, D., AND BEYAH, R. Temporal execution behavior for host anomaly detection in programmable logic controllers. *IEEE Transactions on Information Forensics and Security 15* (2019), 1455–1469.

[70] FORMBY, D., DURBHA, S., AND BEYAH, R. Out of control: Ransomware for industrial control systems. In *RSA conference* (2017), vol. 4.

[71] FORMBY, D., SRINIVASAN, P., LEONARD, A. M., ROGERS, J. D., AND BEYAH, R. A. Who's in control of your control system? device fingerprinting for cyber-physical systems. In *Network and Distributed System Security Symposium* (2016).

[72] FRANCO, J., ARIS, A., CANBERK, B., AND ULUAGAC, A. S. A survey of honeypots and honeynets for internet of things, industrial internet of things, and cyber-physical systems. *IEEE Communications Surveys & Tutorials 23*, 4 (2021), 2351–2383.

[73] GAO, J., LI, J., JIANG, H., LI, Y., AND QUAN, H. A new detection approach against attack/intrusion in measurement and control system with fins protocol. In *2020 Chinese Automation Congress (CAC)* (2020), IEEE, pp. 3691–3696.

[74] GAO, J., LI, Y., LI, Y., GONG, T., XU, F., ZHAO, Q., JIANG, H., AND JIA, J. An effective defense method based on hash authentication against mode-switching attack of ormon plc. In *2022 7th International Conference on Intelligent Computing and Signal Processing (ICSP)* (2022), IEEE, pp. 976–979.

[75] GARCIA, L., BRASSER, F., CINTUGLU, M. H., SADEGHI, A.-R., MOHAMMED, O. A., AND ZONOUZ, S. A. Hey, my malware knows physics! attacking plcs with physical model aware rootkit. In *Network and Distributed System Security Symposium* (2017).

[76] GARCIA, L., SENYONDO, H., MCLAUGHLIN, S., AND ZONOUZ, S. Covert channel communication through physical interdependencies in cyber-physical infrastructures. In *2014 IEEE International Conference on Smart Grid Communications (SmartGridComm)* (2014), IEEE, pp. 952–957.

[77] GARCIA, L., ZONOUZ, S., WEI, D., AND DE AGUIAR, L. P. Detecting plc control corruption via on-device runtime verification. In *2016 Resilience Week (RWS)* (2016), IEEE, pp. 67–72.

[78] GARDINER, J., EIFFERT, A., GARRAGHAN, P., RACE, N., NAGARAJA, S., AND RASHID, A. Controller-in-the-middle: Attacks on software defined networks in industrial control systems. In *Proceedings of the 2th Workshop on CPS&IoT Security and Privacy* (New York, NY, USA, 2021), CPSIoTSec '21, Association for Computing Machinery, p. 63–68.

[79] GHAEINI, H. R., CHAN, M., BAHMANI, R., BRASSER, F., GARCIA, L., ZHOU, J., SADEGHI, A.-R., TIPPENHAUER, N. O., AND ZONOUZ, S. {PAtt}: Physics-based attestation of control systems. In *22nd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2019)* (2019), pp. 165–180.

[80] GIRALDO, J., URBINA, D., CARDENAS, A., VALENTE, J., FAISAL, M., RUTHS, J., TIPPENHAUER, N. O., SANDBERG, H., AND CANDELL, R. A survey of physics-based attack detection in cyber-physical systems. *ACM Comput. Surv. 51*, 4 (jul 2018).

[81] GIRALDO, J., URBINA, D., TANG, C., AND CARDENAS, A. A. The more the merrier: Adding hidden measurements to secure industrial control systems. In *Proceedings of the 7th Symposium on Hot Topics in the Science of Security* (New York, NY, USA, 2020), HotSoS '20, Association for Computing Machinery.

[82] GJENDEMSJØ, M. Creating a weapon of mass disruption: Attacking programmable logic controllers. Master's thesis, Institutt for datateknikk og informasjonsvitenskap, 2013.

[83] GOLDENBERG, N., AND WOOL, A. Accurate modeling of modbus/tcp for intrusion detection in scada systems. *international journal of critical infrastructure protection 6*, 2 (2013), 63–75.

[84] GOVIL, N., AGRAWAL, A., AND TIPPENHAUER, N. O. On ladder logic bombs in industrial control systems. In *Computer Security*. Springer, 2017, pp. 110–126.

[85] GOWDANAKATTE, S., RAY, I., AND HILDE HOUMB, S. Attribute based access control model for protecting programmable logic controllers. In *Proceedings of the 2022 ACM Workshop on Secure and Trustworthy Cyber-Physical Systems* (New York, NY, USA, 2022), Sat-CPS '22, Association for Computing Machinery, p. 47–56.

[86] GRANDGENETT, R., GANDHI, R., AND MAHONEY, W. Exploitation of allen bradley's implementation of ethernet/ip for denial of service against industrial control systems. In *Proceedings of the 9th International Conference on Cyber Warfare and Security, ICCWS 2014, Academic Conferences and Publishing International Limited, Reading, UK* (2014), pp. 58–65.

[87] GRANDGENETT, R., MAHONEY, W., AND GANDHI, R. Authentication bypass and remote escalated i/o command attacks. In *Proceedings of the 10th Annual Cyber and Information Security Research Conference* (2015), pp. 1–7.

[88] GREEN, B., DERBYSHIRE, R., KROTOFIL, M., KNOWLES, W., PRINCE, D., AND SURI, N. Pcaad: Towards automated determination and exploitation of industrial systems. *Computers & Security 110* (2021), 102424.

[89] HADŽIOSMANOVIĆ, D., SOMMER, R., ZAMBON, E., AND 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 York, NY, USA, 2014), ACSAC '14, Association for Computing Machinery, p. 126–135.

[90] HAN, Y., ETIGOWNI, S., LIU, H., ZONOUZ, S., AND PETROPULU, A. Watch me, but don't touch me! contactless control flow monitoring via electromagnetic emanations. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security* (New York, NY, USA, 2017), CCS '17, Association for Computing Machinery, p. 1095–1108.

[91] Hardware configuration changes in RUN mode. https://cache.industry.siemens.com/dl/files/505/14040505/att_47225/v1/cir_en.pdf. Accessed: 02-10-2023.

[92] HASKAMP, H., ORTH, F., WERMANN, J., AND COLOMBO, A. W. Implementing an opc ua interface for legacy plc-based automation systems using the azure cloud: An icps-architecture with a retrofitted rfid system. In *2018 IEEE Industrial Cyber-Physical Systems (ICPS)* (2018), IEEE, pp. 115–121.

[93] HEMSLEY, K. E., FISHER, E., ET AL. History of industrial control system cyber incidents. Tech. rep., Idaho National Lab.(INL), Idaho Falls, ID (United States), 2018.

[94] HILDEBRANDT, M., LAMSHÖFT, K., DITTMANN, J., NEUBERT, T., AND VIELHAUER, C. Information hiding in industrial control systems: An opc ua based supply chain attack and its detection. In *Proceedings of the 2020 ACM Workshop on Information Hiding and Multimedia Security* (New York, NY, USA, 2020), IH&MMSec '20, Association for Computing Machinery, p. 115–120.

[95] HOEVE, M. Detecting intrusions in encrypted control traffic. In *Proceedings of the First ACM Workshop on Smart Energy Grid Security* (Berlin, Germany, 2013), SEGS '13, Association for Computing Machinery, p. 23–28.

[96] HOFFMAN, M., AND CEDILLO, G. Value of PLC Key Switch Monitoring to Keep Critical Systems More Secure | Dragos. https://www.dragos.com/blog/industry-news/value-of-plc-key-switch-monitoring/, Aug. 2021. Accessed: 03-10-2023.

[97] How can you protect your S7 program with a password for an S7-300 and ET 200 CPU? https://support.industry.siemens.com/cs/document/10154913/how-can-you-protect-your-s7-program-with-a-password-for-an-s7-300-and-et-200-cpu-?dti=0&lc=en-US. Accessed: 02-10-2023.

[98] How do you perform a firmware update of the distributed IO in STEP 7 / PCS 7? https://support.industry.siemens.com/cs/document/109783755/how-do-you-perform-a-firmware-update-of-the-distributed-io-in-step-7-pcs-7-?dti=0&pnid=14342&lc=en-US. Accessed: 02-10-2023.

[99] ICS-CERT, T. Mar-17-352-01 hatman-safety system targeted malware (update b). https://www.cisa.gov/sites/default/files/documents/MAR-17-352-01%20HatMan%20-%20Safety%20System%20Targeted%20Malware%20%28Update%20B%29.pdf, Feb. 2019. Accessed: 02-10-2023.

[100] IEEE 802.3 ETHERNET. https://www.ieee802.org/3/. Accessed: 02-10-2023.

[101] IGURE, V. M., LAUGHTER, S. A., AND WILLIAMS, R. D. Security issues in scada networks. *computers & security 25*, 7 (2006), 498–506.

[102] JIN, C., VALIZADEH, S., AND VAN DIJK, M. Snapshotter: Lightweight intrusion detection and prevention system for industrial control systems. In *2018 IEEE Industrial Cyber-Physical Systems (ICPS)* (2018), IEEE, pp. 824–829.

[103] JOSEPHLAL, E. F. M., ADEPU, S., YANG, Z., AND ZHOU, J. Enabling isolation and recovery in plc redundancy framework of metro train systems. *International Journal of Information Security 20*, 6 (2021), 783–795.

[104] JUNJIAO, L., LIN, X., XIN, C., HUI, W., LI, H., YAN, H., JIAWEI, S., ZHIQIANG, S., AND SUN, L. Shadowplcs: A novel scheme for remote detection of industrial process control attacks. *IEEE Transactions on Dependable and Secure Computing* (2020).

[105] KALLE, S., AMEEN, N., YOO, H., AND AHMED, I. Clik on plcs! attacking control logic with decompilation and virtual plc. In *Binary Analysis Research (BAR) Workshop, Network and Distributed System Security Symposium (NDSS)* (2019).

[106] KALOROUMAKIS, P. E., AND SMITH, M. J. Toward a knowledge graph of cybersecurity countermeasures. *The MITRE Corporation* (2021), 11.

[107] KELIRIS, A., AND MANIATAKOS, M. ICSREF: A framework for automated reverse engineering of industrial control systems binaries. In *Network and Distributed System Security Symposium (NDSS)* (2019).

[108] KEREN, T. The Race to Native Code Execution in PLCs. https://claroty.com/blog/research-race-to-native-code-execution-in-plcs. Accessed: 03-10-2023.

[109] KHADPE, M., BINNAR, P., AND KAZI, F. Malware injection in operational technology networks. In *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)* (2020), IEEE, pp. 1–6.

[110] KLEINMAN, A., AND WOOL, A. Accurate modeling of the siemens s7 scada protocol for intrusion detection and digital forensics. *The Journal of Digital Forensics, Security and Law: JDFSL 9*, 2 (2014), 37.

[111] KLICK, J., LAU, S., MARZIN, D., MALCHOW, J.-O., AND ROTH, V. Internet-facing plcs-a new back orifice. *Blackhat USA* (2015), 22–26. https://www.blackhat.com/docs/us-15/materials/us-15-Klick-Internet-Facing-PLCs-A-New-Back-Orifice-wp.pdf. Accessed: 02-10-2023.

[112] KOLLA, S., BORDER, D., AND MAYER, E. Fieldbus networks for control system implementations. In *Proceedings: Electrical Insulation Conference and Electrical Manufacturing and Coil Winding Technology Conference (Cat. No. 03CH37480)* (2003), IEEE, pp. 493–498.

[113] KREIMEL, P., EIGNER, O., AND TAVOLATO, P. Anomaly-based detection and classification of attacks in cyber-physical systems. In *Proceedings of the 12th International Conference on Availability, Reliability and Security* (2017), pp. 1–6.

[114] KRISHNAMURTHY, P., KHORRAMI, F., KARRI, R., PAUL-PENA, D., AND SALEHGHAFFARI, H. Process-aware covert channels using physical instrumentation in cyber-physical systems. *IEEE Transactions on Information Forensics and Security 13*, 11 (2018), 2761–2771.

[115] KROTOFIL, M., AND LARSEN, J. Rocking the pocket book: Hacking chemical plants. *DEFCON Conference* (2017). https://media.defcon.org/DEF%20CON%2023/DEF%20CON%2023%20presentations/DEF%20CON%2023%20-%20Marina-Krotofil-Jason-Larsen-Rocking-the-Pocketbook-Hacking-Chemical-Plants-WP-UPDATED.pdf. Accessed: 02-10-2023.

[116] LADISA, P., PLATE, H., MARTINEZ, M., AND BARAIS, O. Sok: Taxonomy of attacks on open-source software supply chains. In *2023 IEEE Symposium on Security and Privacy (SP)* (2022), IEEE Computer Society, pp. 167–184.

[117] LAN, H., ZHU, X., SUN, J., AND LI, S. Traffic data classification to detect man-in-the-middle attacks in industrial control system. In *2019 6th International Conference on Dependable Systems and Their Applications (DSA)* (2020), IEEE, pp. 430–434.

[118] LANGNER, R. Stuxnet: Dissecting a cyberwarfare weapon. *IEEE Security and Privacy 9*, 3 (may 2011), 49–51.

[119] LEE, J., CHOI, H., SHIN, J., AND SEO, J. T. Detection and analysis technique for manipulation attacks on plc control logic. In *Proceedings of the 2020 ACM International Conference on Intelligent Computing and Its Emerging Applications* (New York, NY, USA, 2021), ACM ICEA '20, Association for Computing Machinery.

[120] LEI, C., DONGHONG, L., AND LIANG, M. The spear to break the security wall of S7CommPlus. *Black Hat Europe* (2017), 12. https://www.blackhat.com/docs/eu-17/materials/eu-17-Lei-The-Spear-To-Break%20-The-Security-Wall-Of-S7CommPlus-wp.pdf. Accessed: 02-10-2023.

[121] LIM, B., CHEN, D., AN, Y., KALBARCZYK, Z., AND IYER, R. Attack induced common-mode failures on plc-based safety system in a nuclear power plant: practical experience report. In *2017 IEEE 22nd Pacific Rim International Symposium on Dependable Computing (PRDC)* (2017), IEEE, pp. 205–210.

[122] LÓPEZ-MORALES, E., RUBIO-MEDRANO, C., DOUPÉ, A., SHOSHITAISHVILI, Y., WANG, R., BAO, T., AND AHN, G.-J. Honeyplc: A next-generation honeypot for industrial control systems. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security* (New York, NY, USA, 2020), CCS '20, Association for Computing Machinery, p. 279–291.

[123] LUO, J., KANG, M., BISSE, E., VELDINK, M., OKUNEV, D., KOLB, S., TYLKA, J. G., AND CANEDO, A. A quad-redundant plc architecture for cyber-resilient industrial control systems. *IEEE Embedded Systems Letters 13*, 4 (2020), 218–221.

[124] MAESSCHALCK, S., STAVES, A., DERBYSHIRE, R., GREEN, B., AND HUTCHINSON, D. Snakes and ladder logic: Plc-vbs, a plc control logic vulnerability discovery tool. *arXiv preprint arXiv:2206.06669* (2022).

[125] MALCHOW, J.-O., MARZIN, D., KLICK, J., KOVACS, R., AND ROTH, V. Plc guard: A practical defense against attacks on cyber-physical systems. In *2015 IEEE Conference on Communications and Network Security (CNS)* (2015), IEEE, pp. 326–334.

[126] MARINA KROTOFIL, R. D. Greetings from the '90s: Exploiting the design of industrial controllers in modern settings. *Black Hat Europe* (2021). https://i.blackhat.com/EU-21/Wednesday/EU-21-Krotofil-Greetings-from-the-90s-Exploiting-the-Design-of-Industrial-Controllers-in-Modern-Settings.pdf. Accessed: 02-10-2023.

[127] MARSHALL, P. S., AND RINALDI, J. S. *Industrial Ethernet*. ISA, 2004.

[128] Matrix | MITRE ATT&CK®. https://attack.mitre.org/matrices/ics/. Accessed: 02-10-2023.

[129] MCLAUGHLIN, S. On dynamic malware payloads aimed at programmable logic controllers. In *Proceedings of the 6th USENIX Conference on Hot Topics in Security* (USA, 2011), HotSec'11, USENIX Association, p. 10.

[130] MCLAUGHLIN, S. Cps: Stateful policy enforcement for control system device usage. In *Proceedings of the 29th Annual Computer Security Applications Conference* (New York, NY, USA, 2013), ACSAC '13, Association for Computing Machinery, p. 109–118.

[131] MCLAUGHLIN, S., AND MCDANIEL, P. Sabot: Specification-based payload generation for programmable logic controllers. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security* (Raleigh, North Carolina, USA, 2012), CCS '12, Association for Computing Machinery, p. 439–449.

[132] MCLAUGHLIN, S., AND ZONOUZ, S. Controller-aware false data injection against programmable logic controllers. In *2014 IEEE International Conference on Smart Grid Communications (SmartGridComm)* (2014), IEEE, pp. 848–853.

[133] MCLAUGHLIN, S. E., ZONOUZ, S. A., POHLY, D. J., AND MCDANIEL, P. D. A trusted safety verifier for process controller code. In *Network and Distributed System Security Symposium* (2014), vol. 14.

[134] MCMINN, L., AND BUTTS, J. A firmware verification tool for programmable logic controllers. In *International Conference on Critical Infrastructure Protection* (2012), Springer, pp. 59–69.

[135] MERA, A., CHEN, Y. H., SUN, R., KIRDA, E., AND LU, L. D-box: Dma-enabled compartmentalization for embedded applications. In *Network and Distributed System Security Symposium* (2022).

[136] MILINKOVIĆ, S. A., AND LAZIĆ, L. R. Industrial plc security issues. In *2012 20th Telecommunications Forum (TELFOR)* (2012), IEEE, pp. 1536–1539.

[137] MITRE ATT&CK®: Design and philosophy. https://attack.mitre.org/docs/ATTACK_Design_and_Philosophy_March_2020.pdf. Accessed: 12-10-2023.

[138] MITRE D3FEND Knowledge Graph. https://d3fend.mitre.org/. Accessed: 02-10-2023.

[139] MORALES, E. L., RUBIO-MEDRANO, C. E., DOUPÉ, A., WANG, R., SHOSHITAISHVILI, Y., BAO, T., AND AHN, G.-J. *HoneyPLC: A Next-Generation Honeypot for Industrial Control Systems*. Springer International Publishing, Cham, 2023, pp. 145–181.

[140] Mqtt: The standard for iot messaging. https://mqtt.org/, 2022. Accessed: 02-10-2023.

[141] MULDER, J., SCHWARTZ, M., BERG, M., VAN HOUTEN, J. R., MARIO, J., URREA, M. A. K., CLEMENTS, A. A., AND JACOB, J. Weaselboard: zero-day exploit detection for programmable logic controllers. *Sandia report SAND2013-8274, Sandia national laboratories* (2013).

[142] MURILLO, A. F., CÓMBITA, L. F., GONZALEZ, A. C., RUEDA, S., CARDENAS, A. A., AND QUIJANO, N. A virtual environment for industrial control systems: A nonlinear use-case in attack detection, identification, and response. In *Proceedings of the 4th Annual Industrial Control System Security Workshop* (San Juan, PR, USA, 2018), ICSS '18, Association for Computing Machinery, p. 25–32.

[143] NARAYAN RAJPUT, P. H., DOUMANIDIS, C., AND MANIATAKOS, M. Icspatch: Automated vulnerability localization and non-intrusive hotpatching in industrial control systems using data dependence graphs. In *Proceedings of the USENIX Security Symposium (USENIX Security)* (2022).

[144] NAUMENKO, N. Simatic mc7 code. https://gitlab.com/nnaumenko/mc7. Accessed: 02-10-2023.

[145] NIEDERMAIER, M., FISCHER, F., AND VON BODISCO, A. Propfuzz—an it-security fuzzing framework for proprietary ics protocols. In *2017 International conference on applied electronics (AE)* (2017), IEEE, pp. 1–4.

[146] NIEDERMAIER, M., MALCHOW, J.-O., FISCHER, F., MARZIN, D., MERLI, D., ROTH, V., AND VON BODISCO, A. You snooze, you lose: measuring {PLC} cycle times under attacks. In *12th USENIX Workshop on Offensive Technologies (WOOT 18)* (2018).

[147] 1.1 OpenPLC Overview – OpenPLC. https://openplcproject.com/docs/openplc-overview/. Accessed: 27-09-2023.

[148] OR-MEIR, O., NISSIM, N., ELOVICI, Y., AND ROKACH, L. Dynamic malware analysis in the modern era—a state of the art survey. *ACM Comput. Surv. 52*, 5 (sep 2019).

[149] PEARCE, H., PINISETTY, S., ROOP, P. S., KUO, M. M., AND UKIL, A. Smart i/o modules for mitigating cyber-physical attacks on industrial control systems. *IEEE Transactions on Industrial Informatics 16*, 7 (2019), 4659–4669.

[150] PECK, D., AND PETERSON, D. Leveraging ethernet card vulnerabilities in field devices. In *SCADA security scientific symposium* (2009), pp. 1–19.

[151] PICKREN, R., SHEKARI, T., ZONOUZ, S., AND BEYAH, R. Compromising industrial processes using web-based programmable logic controller malware. In *Network and Distributed System Security Symposium (NDSS)* (2024).

[152] PLESOWICZ, P., AND LASZCZYK, P. Evaluation of secure signal transmission in automatic control using ssh tunneling. In *2013 18th International Conference on Methods & Models in Automation & Robotics (MMAR)* (2013), IEEE, pp. 672–677.

[153] PONOMAREV, S., AND ATKISON, T. Industrial control system network intrusion detection by telemetry analysis. *IEEE Transactions on Dependable and Secure Computing 13*, 2 (2015), 252–260.

[154] QASIM, S. A., AYUB, A., JOHNSON, J., AND AHMED, I. Attacking the iec 61131 logic engine in programmable logic controllers. In *International Conference on Critical Infrastructure Protection* (2021), Springer, pp. 73–95.

[155] QASIM, S. A., LOPEZ, J., AND AHMED, I. Automated reconstruction of control logic for programmable logic controller forensics. In *International Conference on Information Security* (2019), Springer, pp. 402–422.

[156] RAJPUT, P. H. N., SARKAR, E., TYCHALAS, D., AND MANIATAKOS, M. Remote non-intrusive malware detection for plcs based on chain of trust rooted in hardware. In *2021 IEEE European Symposium on Security and Privacy (EuroS&P)* (2021), IEEE, pp. 369–384.

[157] ROBLES-DURAZNO, A., MORADPOOR, N., MCWHINNIE, J., AND RUSSELL, G. Waterleakage: A stealthy malware for data exfiltration on industrial control systems using visual channels. In *2019 IEEE 15th International Conference on Control and Automation (ICCA)* (2019), IEEE, pp. 724–731.

[158] ROBLES-DURAZNO, A., MORADPOOR, N., MCWHINNIE, J., RUSSELL, G., AND MANERU-MARIN, I. Plc memory attack detection and response in a clean water supply system. *International Journal of Critical Infrastructure Protection 26* (2019), 100300.

[159] SAARIMÄKI, V., SILTALA, N., PARTANEN, P., VIHINEN, J., AND TUOKKO, R. Experiences in different fieldbuses used together with pc-based control systems. In *Fieldbus Technology*. Springer, 1999, pp. 164–169.

[160] SALEHI, M., AND BAYAT-SARMADI, S. Plcdefender: Improving remote attestation techniques for plcs using physical model. *IEEE Internet of Things Journal 8*, 9 (2020), 7372–7379.

[161] SANTAMARTA, R. Attacking controllogix. https://drive.google.com/file/d/1Ch_1PvDYd1QYcdhZr9NhtpuSGcbnhW1v/, 2012. Accessed: 27-09-2023.

[162] SAPIR, M., KATZ, U., MOSHE, N., BRIZINOV, S., AND PREMINGER, A. EVIL PLC ATTACK: WEAPONIZING PLCS. *Team82 Research* (2022), 60. https://claroty-statamic-assets.nyc3.digitaloceanspaces.com/resource-downloads/team82-evil-plc-attack-research-paper-1661285586.pdf. Accessed: 27-09-2023.

[163] SARKAR, E., BENKRAOUDA, H., AND MANIATAKOS, M. I came, i saw, i hacked: Automated generation of process-independent attacks for industrial control systems. In *Proceedings of the 15th ACM Asia Conference on Computer and Communications Security* (New York, NY, USA, 2020), ASIA CCS '20, Association for Computing Machinery, p. 744–758.

[164] SASAKI, T., FUJITA, A., GAÑÁN, C. H., VAN EETEN, M., YOSHIOKA, K., AND MATSUMOTO, T. Exposed infrastructures: Discovery, attacks and remediation of insecure ics remote management devices. In *2022 IEEE Symposium on Security and Privacy (SP)* (2022), pp. 2379–2396.

[165] SAYEGH, N., CHEHAB, A., ELHAJJ, I. H., AND KAYSSI, A. Internal security attacks on scada systems. In *2013 Third International Conference on Communications and Information Technology (ICCIT)* (2013), IEEE, pp. 22–27.

[166] SCHORRADT, S., BAJRAMOVIC, E., AND FREILING, F. On the feasibility of secure logging for industrial control systems using blockchain. In *Proceedings of the Third Central European Cybersecurity Conference* (New York, NY, USA, 2019), CECC 2019, Association for Computing Machinery.

[167] SCHUETT, C., BUTTS, J., AND DUNLAP, S. An evaluation of modification attacks on programmable logic controllers. *International Journal of Critical Infrastructure Protection 7*, 1 (2014), 61–68.

[168] SEGOVIA, V. R., AND THEORIN, A. History of control history of plc and dcs. *University of Lund* (2012).

[169] SENTHIVEL, S., DHUNGANA, S., YOO, H., AHMED, I., AND ROUSSEV, V. Denial of engineering operations attacks in industrial control systems. In *Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy* (New York, NY, USA, 2018), CODASPY '18, Association for Computing Machinery, p. 319–329.

[170] SENYONDO, H., SUN, P., BERTHIER, R., AND ZONOUZ, S. Plcloud: Comprehensive power grid plc security monitoring with zero safety disruption. In *2015 IEEE International Conference on Smart Grid Communications (SmartGridComm)* (2015), IEEE, pp. 809–816.

[171] SHANG, W., QIAO, Q., WAN, M., AND ZENG, P. Design and implementation of industrial firewall for modbus/tcp. *J. Comput. 11*, 5 (2016), 432–438.

[172] SHRIVASTAVA, S., MR, G. R., AND MATHUR, A. Pcat: Plc command analysis tool for automatic incidence response in water treatment plants. In *2021 IEEE International Conference on Big Data (Big Data)* (2021), IEEE, pp. 2151–2159.

[173] SIEMENS. The intelligent choice for your automation task: Simatic controllers. Accessed: 2021-12-8.

[174] Siemens chooses vxworks to power its high-performance industrial pcs. https://www.windriver.com/themes/Windriver/pdf/Siemens-Customer-Success-Story.pdf. Accessed: 03-10-2023.

[175] Siemens embedded software board support packages. https://static.sw.cdn.siemens.com/siemens-disw-assets/public/5TLWV4zZ3yuS84kwVz3Tqb/en-US/Siemens_SW_embedded_board_support_packages_v9_Nov2022.pdf, Nov. 2022. Accessed: 02-10-2023.

[176] SISINNI, E., SAIFULLAH, A., HAN, S., JENNEHAG, U., AND GIDLUND, M. Industrial internet of things: Challenges, opportunities, and directions. *IEEE transactions on industrial informatics 14*, 11 (2018), 4724–4734.

[177] SLOWIK, J. Anatomy of an attack: Detecting and defeating crashoverride. *VB2018, October* (2018).

[178] Soft PLCs: Revisiting the industrial innovator's dilemma. https://iot-analytics.com/software-based-plcs-revisiting-industrial-innovators-dilemma/, Oct. 2021. Accessed: 03-10-2023.

[179] SoftPLC. https://en.everybodywiki.com/SoftPLC, Aug. 2020. Accessed: 03-10-2023.

[180] SPENNEBERG, R., BRÜGGEMANN, M., AND SCHWARTKE, H. Plc-blaster: A worm living solely in the plc. *Black Hat Asia 16* (2016), 1–16. https://www.blackhat.com/docs/asia-16/materials/asia-16-Spenneberg-PLC-Blaster-A-Worm-Living-Solely-In-The-PLC-wp.pdf. Accessed: 03-10-2023.

[181] STANKOVIC, J. A., AND RAJKUMAR, R. Real-time operating systems. *Real-Time Systems 28*, 2-3 (2004), 237–253.

[182] STOUFFER, K., FALCO, J., AND SCARFONE, K. Guide to industrial control systems (ics) security - supervisory control and data acquisition (scada) systems, distributed control systems (dcs), and other control system configurations such as programmable logic controllers (plc), 2011-06-07 2011. https://www.nist.gov/publications/guide-industrial-control-systems-ics-security-supervisory-control-and-data-acquisition. Accessed: 27-09-2023.

[183] STOUFFER, K., FALCO, J., AND SCARFONE, K. Guide to industrial control systems (ics) security–rev. 2. *NIST Special Publication 800*, 82 (2015).

[184] STOUFFER, K., PEASE, M., TANG, C., ZIMMERMAN, T., PILLITTERI, V., AND LIGHTMAN, S. Guide to operational technology (ot) security. *National Institute of Standards and Technology: Gaithersburg, MD, USA NIST SP 800-82 Rev. 3* (Sept. 2023).

[185] SUN, R., MERA, A., LU, L., AND CHOFFNES, D. Sok: Attacks on industrial control logic and formal verification-based defenses. In *2021 IEEE European Symposium on Security and Privacy (EuroS&P)* (2021), IEEE, pp. 385–402.

[186] System firmware, technique t0857 - ics | mitre att&ck®. https://attack.mitre.org/techniques/T0857/. Accessed: 04-11-2023.

[187] THOMESSE, J.-P. Fieldbus technology in industrial automation. *Proceedings of the IEEE 93*, 6 (2005), 1073–1101.

[188] TIEGELKAMP, M., AND JOHN, K.-H. *IEC 61131-3: Programming industrial automation systems*. Springer, 2010.

[189] TIPPENHAUER, N. O., CHEN, B., MASHIMA, D., AND NICOL, D. M. Vbump: Securing ethernet-based industrial control system networks with vlan-based traffic aggregation. In *Proceedings of the 2th Workshop on CPS&IoT Security and Privacy* (Virtual Event, Republic of Korea, 2021), CPSIoTSec '21, Association for Computing Machinery, p. 3–14.

[190] Top 20 Secure PLC Coding Practices. https://plc-security.com/. Accessed: 03-10-2023.

[191] TYCHALAS, D., BENKRAOUDA, H., AND MANIATAKOS, M. Icsfuzz: Manipulating i/os and repurposing binary code to enable instrumented fuzzing in ics control applications. In *30th USENIX Security Symposium (USENIX Security 21)* (Virtual Event, 2021).

[192] TYCHALAS, D., KELIRIS, A., AND MANIATAKOS, M. Stealthy information leakage through peripheral exploitation in modern embedded systems. *IEEE Transactions on Device and Materials Reliability 20*, 2 (2020), 308–318.

[193] TYCHALAS, D., AND MANIATAKOS, M. Special session: Potentially leaky controller: Examining cache side-channel attacks in programmable logic controllers. In *2020 IEEE 38th International Conference on Computer Design (ICCD)* (2020), IEEE, pp. 33–36.

[194] Unified Architecture - opc foundation. https://opcfoundation.org/about/opc-technologies/opc-ua/. Accessed: 03-10-2023.

[195] URBINA, D., GIRALDO, J., TIPPENHAUER, N. O., AND CARDENAS, A. Attacking fieldbus communications in ics: Applications to the swat testbed. In *Proceedings of the Singapore Cyber-Security Conference (SG-CRC) 2016* (2016), IOS Press, pp. 75–89.

[196] URBINA, D. I., GIRALDO, J. A., CARDENAS, A. A., TIPPENHAUER, N. O., VALENTE, J., FAISAL, M., RUTHS, J., CANDELL, R., AND SANDBERG, H. Limiting the impact of stealthy attacks on industrial control systems. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (Vienna, Austria, 2016), CCS '16, Association for Computing Machinery, p. 1092–1105.

[197] VIDAS, T., AND CHRISTIN, N. Evading android runtime analysis via sandbox detection. In *Proceedings of the 9th ACM Symposium on Information, Computer and Communications Security* (Kyoto, Japan, 2014), ASIA CCS '14, Association for Computing Machinery, p. 447–458.

[198] Vulnerability disclosure policy template | cisa. https://www.cisa.gov/vulnerability-disclosure-policy-template. Accessed: 14-11-2023.

[199] WALITA, T., ERBA, A., CASTELLANOS, J. H., AND TIPPENHAUER, N. O. Blind concealment from reconstruction-based attack detectors for industrial control systems via backdoor attacks. In *Proceedings of the 9th ACM Cyber-Physical System Security Workshop* (Melbourne, VIC, Australia, 2023), CPSS '23, Association for Computing Machinery, p. 36–47.

[200] WALKUP, G., ETIGOWNI, S., XU, D., URIAS, V., AND LIN, H. W. Forensic investigation of industrial control systems using deterministic replay. In *2020 IEEE Conference on Communications and Network Security (CNS)* (2020), IEEE, pp. 1–9.

[201] WANG, T., XIONG, Q., GAO, H., PENG, Y., DAI, Z., AND YI, S. Design and implementation of fuzzing technology for opc protocol. In *2013 Ninth International Conference on Intelligent Information Hiding and Multimedia Signal Processing* (2013), IEEE, pp. 424–428.

[202] WARDAK, H., ZHIOUA, S., AND ALMULHEM, A. Plc access control: a security analysis. In *2016 World Congress on Industrial Control Systems Security (WCICSS)* (2016), IEEE, pp. 1–6.

[203] WEDGBURY, A., AND JONES, K. Automated asset discovery in industrial control systems-exploring the problem. In *3rd International Symposium for ICS & SCADA Cyber Security Research 2015 (ICS-CSR 2015) 3* (2015), pp. 73–83.

[204] WIGHTMAN, R., AND PETERSON, D. Project Basecamp 3S CoDeSys Vulns and Tools. https://web.archive.org/web/20130806010310/http:/www.digitalbond.com/tools/basecamp/3s-codesys/, Aug. 2013. Accessed: 03-10-2023.

[205] WILLIAMS, T. J. The purdue enterprise reference architecture. *Computers in industry 24*, 2-3 (1994), 141–158.

[206] WOHLIN, C. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering* (London, England, United Kingdom, 2014), EASE '14, Association for Computing Machinery.

[207] XIAO, M., WU, J., LONG, C., AND LI, S. Construction of false sequence attack against plc based power control system. In *2016 35th Chinese Control Conference (CCC)* (2016), IEEE, pp. 10090–10095.

[208] XU, Y., YANG, Y., LI, T., JU, J., AND WANG, Q. Review on cyber vulnerabilities of communication protocols in industrial control systems. In *2017 IEEE Conference on Energy Internet and Energy System Integration (EI2)* (2017), IEEE, pp. 1–6.

[209] YANG, Z., BAO, Z., JIN, C., LIU, Z., AND ZHOU, J. Plcrypto: A symmetric cryptographic library for programmable logic controllers. *IACR Transactions on Symmetric Cryptology* (2021), 170–217.

[210] YANG, Z., HE, L., CHENG, P., CHEN, J., YAU, D. K., AND DU, L. {PLC-Sleuth}: Detecting and localizing {PLC} intrusions using control invariants. In *23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2020)* (2020), pp. 333–348.

[211] YAU, K., AND CHOW, K.-P. Plc forensics based on control program logic change detection. *Journal of Digital Forensics, Security and Law 10*, 4 (2015), 5.

[212] YLMAZ, E. N., CIYLAN, B., GÖNEN, S., SINDIREN, E., AND KARACAYILMAZ, G. Cyber security in industrial control systems: Analysis of dos attacks against plcs and the insider effect. In *2018 6th international istanbul smart grids and cities congress and fair (icsg)* (2018), IEEE, pp. 81–85.

[213] YOO, H., AND AHMED, I. Control logic injection attacks on industrial control systems. In *IFIP International Conference on ICT Systems Security and Privacy Protection* (2019), Springer, pp. 33–48.

[214] YOO, H., KALLE, S., SMITH, J., AND AHMED, I. Overshadow plc to detect remote control-logic injection attacks. In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment* (2019), Springer, pp. 109–132.

[215] ZADDACH, J., AND COSTIN, A. Embedded devices security and firmware reverse engineering. *Black Hat USA* (2013). https://media.blackhat.com/us-13/US-13-Zaddach-Workshop-on-Embedded-Devices-Security-and-Firmware-Reverse-Engineering-Slides.pdf. Accessed: 03-10-2023.

[216] ZHANG, M., CHEN, C.-Y., KAO, B.-C., QAMSANE, Y., SHAO, Y., LIN, Y., SHI, E., MOHAN, S., BARTON, K., MOYNE, J., ET AL. Towards automated safety vetting of plc code in real-world plants. In *2019 IEEE Symposium on Security and Privacy (SP)* (2019), IEEE, pp. 522–538.

[217] ZHANG, W., JIAO, Y., WU, D., SRINIVASA, S., DE, A., GHOSH, S., AND LIU, P. Armor plc: A platform for cyber security threats assessments for plcs. *Procedia Manufacturing 39* (2019), 270–278.

[218] ZHANG, Y., SUN, Z., YANG, L., LI, Z., ZENG, Q., HE, Y., AND ZHANG, X. All your plcs belong to me: Ics ransomware is realistic. In *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)* (2020), IEEE, pp. 502–509.

[219] ZHOU, R., WANG, Z., JI, X., AND XU, W. Anomoly detection for plc based on magnetic side channel. In *2020 IEEE 4th Conference on Energy Internet and Energy System Integration (EI2)* (2020), IEEE, pp. 3817–3821.

[220] ZUBAIR, N., AYUB, A., YOO, H., AND AHMED, I. Control logic obfuscation attack in industrial control systems. In *2022 IEEE International Conference on Cyber Security and Resilience (CSR)* (2022), IEEE, pp. 227–232.

[221] ZUBAIR, N., AYUB, A., YOO, H., AND AHMED, I. Pem: Remote forensic acquisition of plc memory in industrial control systems. *Forensic Science International: Digital Investigation 40* (2022), 301336.

Appendices

Appendix A Scientific and Grey Literature Resources

The seven digital libraries queried during our scientific literature review are as follows: ACM Digital Library[7], arXiv[8], dblp[9], Google Scholar[10], IEEExplore[11], USENIX Papers Search[12], and NDSS Symposium Search[13].

The three main sources we queried during our grey literature review are as follows: Digital Bond Archives[14], InfoconDB[15], and Google[16]

Search keywords.

```
"plc", "programmable logic controller", "scada", "cps",
"cyber-physical system", "cyber physical system", "iiot",
"industrial internet of things", "industry 4.0",
"industrial control system", "ics", "embedded system",
"attack", "threat", "vulnerability", "defense"
```

Appendix B MITRE Technique and Sub-Technique Model Example

As an example of how our ICS^2 Matrix can be used to categorize ICS-specific attacks proposed by researchers, we will have a look at the work by *Krishnamurthy et al.* [114] They proposed an attack strategy where a malware-infected PLC can exploit the acoustic emissions generated by a motor controlling a valve in a feedback control loop within an ICS. This covert acoustic channel enables the malware to secretly transmit sensitive information, such as proprietary controller parameters and system passwords, to a remote receiver. This attack does not disrupt the stability, performance, or signal characteristics of the closed-loop process, making it stealthy and effective at exfiltrating data from the compromised ICS. We categorized this attack under the *Exfiltration* tactic but could not find a fitting technique. Thus, we propose the addition of the technique *Exfiltration over a Covert Channel*. This way of data exfiltration bypasses the techniques *Filter network Traffic* and *Data Loss Prevention*, as covert channels utilize unintended means of communication by definition. To mitigate covert channels requiring the propagation of wireless signals, *Minimize Wireless Signal Propagation* represents a defense, but due to the different forms that covert channels can take, complete mitigation proves difficult, which puts this technique in the category *Mitigation Limited or Not Effective*. Table 3 shows the new technique definition motivated by the example above. This technique definition follows MITRE's official guidelines [137].

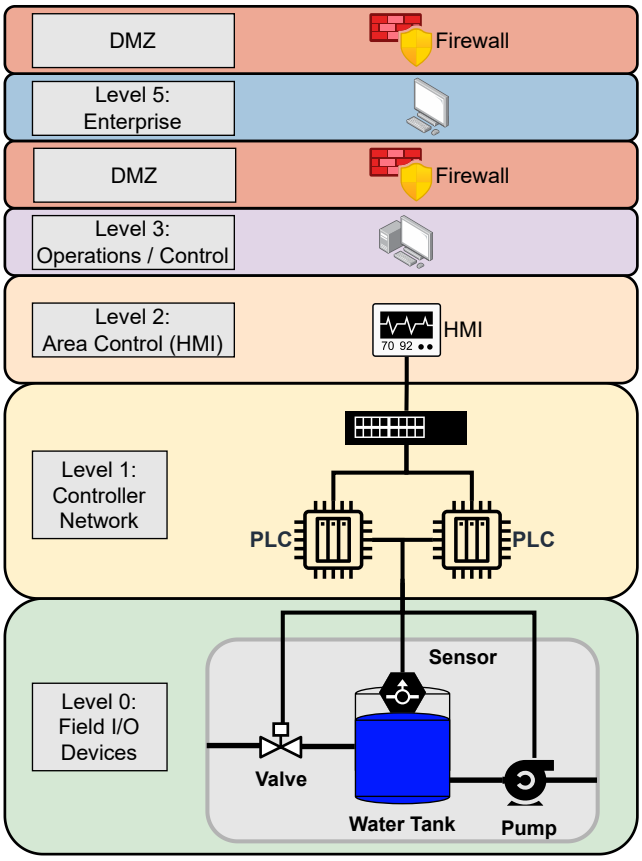

Figure 11: Example of an ICS process with PLCs in Level 1: Controller Network as defined by the Purdue Model. Based on [80, 184].

Appendix C PLCs' Larger Context and Underlying Architecture

In this Appendix, we expand on the environment in which PLCs operate. Specifically, we use the Purdue model [205] to describe the network architecture of an ICS process and where PLCs fit in.

As we discuss in Sec. 1, PLCs control physical machines or actuators, such as pumps. However, these actuators do not exist in a vacuum. They are one component of many that work together to complete a larger industrial process, for example, a water treatment process. Figure 11 depicts a sample of a water treatment process ICS using the Purdue Model.

[7] https://dl.acm.org/
[8] https://arxiv.org/
[9] https://dblp.org/
[10] https://scholar.google.com/
[11] https://ieeexplore.ieee.org/Xplore/home.jsp
[12] https://www.usenix.org/publications/proceedings/
[13] https://www.ndss-symposium.org/
[14] https://dale-peterson.com/digital-bond-archives/
[15] https://infocondb.org/
[16] https://www.google.com/

Data Item	
Name	Exfiltration over Covert Channel
Tactic	Exfiltration
Data Sources	
Description	Adversaries may attempt to exfiltrate data via a covert channel such as analog emissions of physical instrumentation. For example, actuators, sensors, and mechanical structures. These analog emissions can be acoustic or electromagnetic. In some circumstances, the adversary needs to have physical access to the ICS to measure the analog signal using an antenna, for example. The physical medium or device could be used as the final exfiltration point or to hop between otherwise disconnected systems.
Asset	Field Controller/RTU/PLC/IED
Defense Bypassed	Data Loss Prevention, Filter Network Traffic
Contributor	Anonymized
Procedure Example	Krishnamurthy, Prashanth, et al. "Process-aware covert channels using physical instrumentation in cyber-physical systems." [114] IEEE Transactions on Information Forensics and Security 13.11 (2018): 2761-2771.
Mitigation	Minimize Wireless Signal Propagation, Mitigation Limited or Not Effective
Detection	

Table 3: Proposed technique definition of Exfiltration over Covert Channel according to MITRE ATT&CK: Design and Philosophy [137]

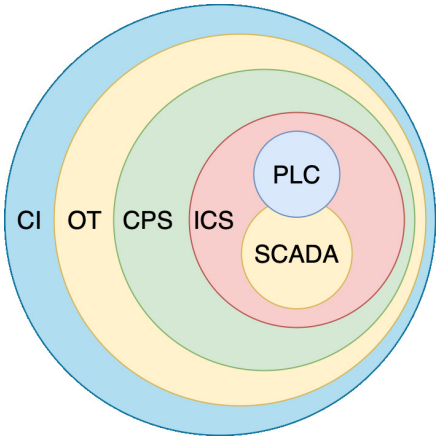

Figure 12: Relationship between Critical Infrastructure (CI), Operational Technology (OT), Cyber-Physical Systems (CPS), Industrial Control Systems (ICS), Supervisory Control and Data Acquisition (SCADA), and PLCs. Based on [139]

Appendix D PLC Memory Blocks

PLCs have specialized memory blocks that store different types of data. These blocks are made available to applications that perform operations like program upload and download [36] and can be classified as:

Data Blocks (DB). These blocks are used to store data that will later be used by a program. Different data types can be stored (e.g., Boolean, byte, integer) [7].

System Data Blocks (SDB). These blocks contain PLC configuration information [7,65], listing PLC model, firmware version, IP address, and information about the attached add-ons (e.g., communication processors, frequency converters). SDBs are automatically created and compiled by the PLC and cannot be modified by the user.

Organization Blocks (OB). These blocks are the interfaces between the operating system and the user program [6, 120]. OBs execute when events occur (e.g., at CPU startup, clocked executions, errors, hardware interrupts). Several other OBs serve specific roles.

Function Blocks (FB). Finally, FBs hold standard executable code blocks, which are written in any of the IEC 61131 standard programming languages [7, 65].

Appendix E PLC Built-in Security Features

Access Control. Most PLCs include built-in access control features. For example, Siemens and Allen-Bradley PLCs can be configured with a password that restricts changes to the CPU configuration [23, 97]. However, these built-in features have been shown to be ineffective [202].

Encryption. Some PLCs also include encryption features for memory blocks, for example, Siemens allows the encryption of specific memory blocks [60]. Another encryption functionality involves the use of hashing to detect changes. Some Rockwell and Siemens PLCs support hashing [190]. Additionally, some of the PLC Industrial Ethernet protocols incorporate built-in encryption. However, like the access control features, they have been shown to be vulnerable [120].

Operational Modes. PLCs have different operational modes intended for different scenarios. For example, to modify the control logic program of a production PLC, it is necessary to change or *switch* the PLC operational mode from *Run* to *Stop* and then to *Program* [96]. In practice, there exist two ways to switch the operational mode of a PLC: either to use the PLC management software, e.g., SIMATIC STEP

7 for Siemens PLCs, or to manually switch the operational mode using a physical *key* inserted in the PLC chassis, which overrides the software option just described.

Overall, the most common operational modes are:

- *Run Mode.* This mode is used to execute the control logic program of the PLC. Input from sensor devices is monitored, and output is sent to actuator devices. Ideally, a production PLC should always be set to *Run* [49, 96].
- *Stop Mode.* In this mode, the PLC stops reading inputs and stops the control logic program execution. Typically, a PLC must be switched to *Stop* before it can be re-configured by means of the *Program* mode [91, 96].
- *Program Mode.* In this mode, the PLC control logic program is loaded, modified, or deleted. While in this mode, all outputs from the PLC are stopped [49, 96].

Disabling Unused Protocols. Some PLCs have the option to disable network services to reduce unnecessary attack surfaces. For example, Siemens PLCs allow for the integrated web server to be disabled [64], preventing the exploitation of web server-based vulnerabilities [31, 136].

Monitoring HMI Data. HMIs are useful to visualize trends about the performance of PLCs, as they can plot the PLC Scan Cycle, the uptime, and shut down and restarts, which can be helpful to detect an ongoing attack [190].

Access Level	Vector
AL3: Physical	• JTAG Port • USB Port • Key Switch • Backplane • Memory Card
AL2: Fieldbus	• PROFIBUS • DeviceNet • RS-232 Serial • EtherNet/IP
AL1: LAN	• GOOSE • Workstation • SoMachine (Schneider) • TriStation (Schneider) • LS Proprietary (LS Electric)
AL0: Internet	• Modbus TCP • DNP3 • IEC 104 • OPC UA • EtherNet/IP • S7comm • Webserver (HTTP/HTTPS) • SNMP

Table 4: Attack Vectors per Access Level.

Appendix F Real-World PLC Attacks

We summarize real-world attacks that showcase the pressing need to improve the security of PLCs.

Stuxnet. Stuxnet is the first-ever documented malware for PLCs. At the time, it set itself apart from previous malware by showing a high level of sophistication, a deep understanding of industrial processes, and the use of four *zero-day* exploits [65]. After compromising a computer with the software for programming PLCs, the malware uploaded its malicious control program to the target PLCs. In particular, the attack targeted Siemens 315 and 417 and made them damage centrifuges while reporting that everything was normal [118].

Triton. This malware, also known as TRISIS and HatMan [56, 99] was identified in 2017 after a petrochemical facility in Saudi Arabia was shut down. After compromising an engineering workstation, Triton was able to launch a dropper (trilog.exe) to deliver backdoor files to Safety Instrumented System (SIS) PLC. The first backdoor file was a 0-day exploit that allowed the attackers to inject the second file into the PLC's memory. With a program in the memory of the PLC, the attackers could have control of the device. The attackers were unable to take full control of the system because an error in the PLC caused a system shutdown.

Pipedream Toolkit. At the time of writing, Pipedream (also known as Incontroller) is the latest documented malware that specifically targets PLCs [57]. This is not a single purpose malware but a modular framework that includes multiple exploits that target different PLCs. Once the attackers compromise a computer in the control network, Pipedream can be used to scan and compromise Scheider Electric PLCs, OMROM Sysmac NEX PLCs, and Open Platform Communications Unified Architecture (OPC UA) servers. Pipedream is believed to have been developed by a nation state [52].

Crashoverride. Also known as Industroyer [177], Crashoverride is believed to have caused the power outage in Ukraine's capital in December of 2016 [93]. A sophisticated malware designed to disrupt ICS networks used in electrical substations, it targeted the Open Platform Communications Data Access protocol or OPC-DA for short, which defines how data can be transferred to and from PLCs.

Appendix G Overview of Defenses

Defenses Focusing on Network Inspection. Since most of the attacks in the literature focus on network exploits, it is natural to expect several defenses in the network as well. Some of the first and most popular network intrusion detection systems proposed for industrial networks model the highly-periodic network traffic as a deterministic finite automaton (DFA) and propose to raise alerts when the network traffic does not follow the learned DFA [110]. Other approaches for protecting the networks of PLCs include adding cryptographic protections and machine learning for anomaly detection and prevention [20].

Control Logic Defenses. Ensuring that the control program running on the PLC is verified and correct is an old area of research, as it is not necessarily related to security but

focuses on providing safety guarantees for the operation of a process [33]. This area has received renewed attention in the security community. An example of this line of research is illustrated by *McLaughlin et al.* [133], which proposed a Trusted Safety Verifier approach that verifies the control logic code through a bump-in-the-wire before it is executed by the PLC.

CPU Defenses. One way to detect attacks against the CPU module is to monitor all communications being exchanged between the modules of a PLC. WeaselBoard [141] is a backplane analysis system that forwards all inter-module traffic to an analysis system to detect a variety of attacks. This approach is more general than just detecting CPU attacks, but because it captures information directly from the CPU, we think it fits better in this category. Other papers on CPU defenses focus on resiliency. For example, *Luo et al.* [123] proposed Quad-Redundand PLC, a redundancy framework to provide resiliency after one CPU is attacked, aiming for a second CPU to keep the system running.

Firmware Modification Defenses. Bump-in-the-wire defenses are general mechanisms where an extra device is placed between the sender and the receiver. This device checks that the data or code sent to the receiver is correct. Some of the more popular firmware defenses are based on this architecture, where the proxy device is used to inspect firmware updates before they are installed in the PLC [30, 134].

I/O Defenses. The two most-cited defenses against the inputs and outputs of the PLC include a defense strategy [196] and a prevention mechanism [130]. The work of Urbina et al. [196] focuses on using physical models of the process under control to detect inconsistencies between the inputs and outputs. On the other hand, C^2 [130] focuses on preventing malicious outputs from a compromised PLC from reaching actuators. C^2 proposes a way to express and enforce security policies, and it also denies actions that violate the policy.

Runtime Defenses. ECFI [3] is a control-flow integrity monitoring runtime protection for real-time PLCs. Fundamentally, ECFI achieves this by segregating control-flow verification of the PLC runtime software from control-flow tracing, ensuring the preservation of real-time requirements crucial for PLC operations. Distinguishing itself from other control-flow monitoring systems, ECFI refrains from terminating the runtime process; instead, it promptly notifies the operator of any detected control-flow violations. This approach strikes a balance between protection and timeliness, allowing for effective threat detection without disrupting the overall system functionality. On the other hand, Ghostbuster [1] serves as a defense mechanism designed to identify Pin Control Attacks targeting PLC runtimes [2]. Implemented as a kernel driver, Ghostbuster operates in two distinct modes to ensure comprehensive protection. In Kernel mode, it actively detects alterations made to the SoC debug registers, enabling the detection of elusive "Ghost in the PLCs" I/O intercepts. On the other hand, to thwart user-mode pin control attacks, Ghostbuster employs a distinct strategy. It diligently monitors the SoC configuration of the Pin Control Subsystem, constantly searching for configuration violations that could sever the connection between the PLC runtime and the physical world. By adopting these proactive measures, Ghostbuster effectively fortifies PLC systems against Pin Control Attacks and reinforces the security of the runtime environment.

OS Defenses. Another IT protection being considered is malware detection at the OS level [156], which requires new hardware (Power Debug PRO) and could be circumvented using data hooking or gaining kernel privilege level.

Memory Defenses. Applying protections like ASLR is now common in most IT infrastructure, but similar approaches are being explored for PLCs. For example, *Robles-Durazno et al.* [158] proposed Optimized Datablocks, an approach in which the allocation of the datablock data is randomized such that the attacker does not know its exact location, making attacks more difficult as with other moving target defenses.

Appendix H Excerpt of the ICS^2 Matrix

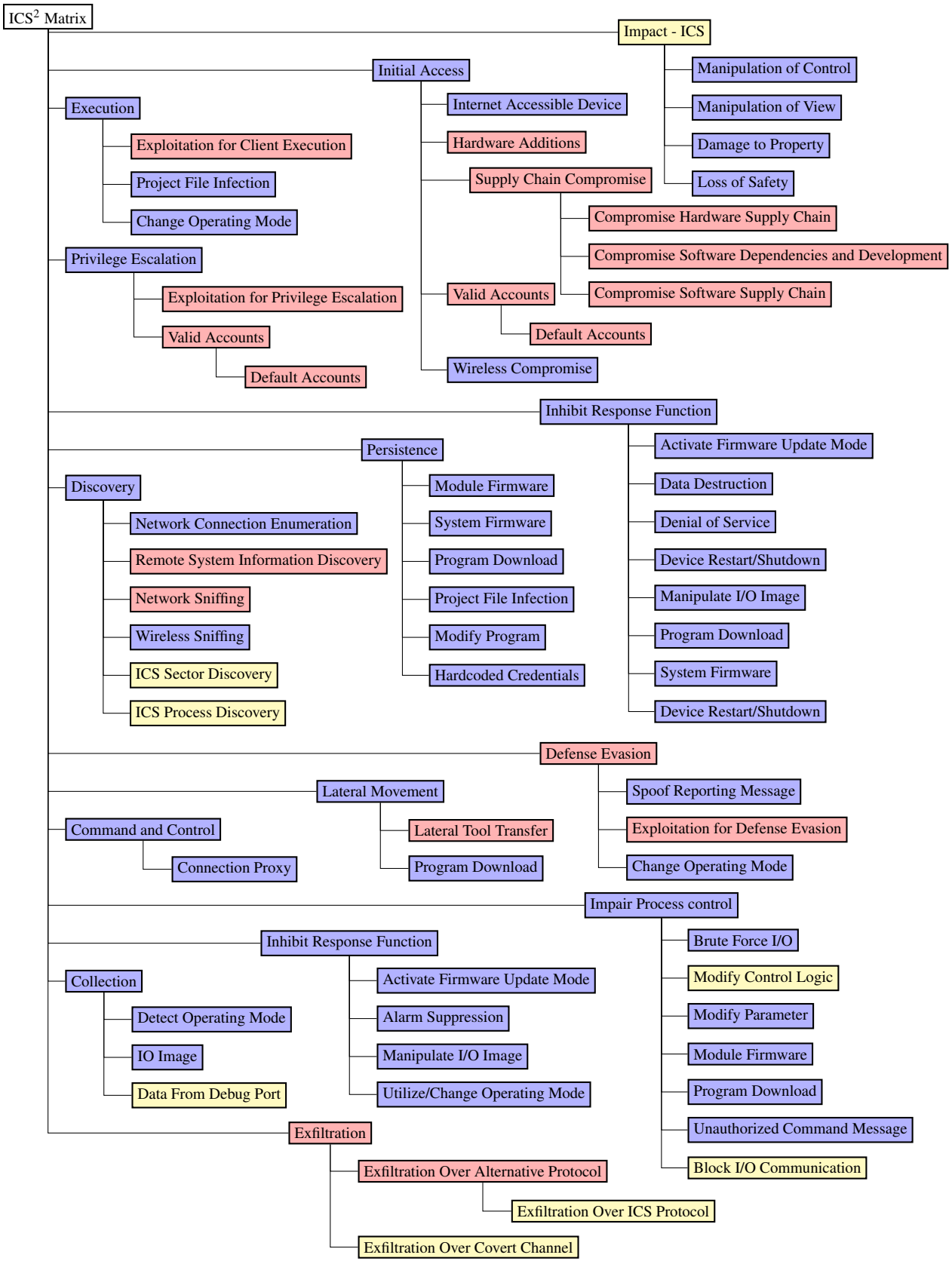

Figure 13: Condensed version of the ICS² Matrix. It includes nodes from the MITRE ATT&CK for ICS Matrix (Blue), the MITRE ATT&CK Enterprise (Red) and our additions (Yellow)