Programmable Logic Controllers - Insecure by Design? A Survey.

Benedikt Geisler
Faculty of Computer Science and Mathematics
Ostbayerische Technische Hochschule Regensburg
Regensburg, Germany
email: benedikt1.geisler@st.oth-regensburg.de

Markus Kucera
Faculty of Computer Science and Mathematics
Ostbayerische Technische Hochschule Regensburg
Regensburg, Germany
email: markus.kucera@oth-regensburg.de

***Abstract*—Any cyber-physical system, including critical infrastructure such as a smart grid, is very likely being controlled by an Industrial Control System (ICS). However, ICSs have long been neglected in terms of security mechanisms. This work presents an overview of the current situation by conducting a literature review, focusing on attack vectors against Programmable Logic Controllers (PLC). Due to proprietary protocols and operating systems, it will present attacks against four major vendors: Siemens, Allen Bradley, Schneider, and Beckhoff.**

***Index Terms*—OT, PLC, ICS, Modbus, Profinet**

I. Introduction

Stuxnet [1], destroying almost 1000 centrifuges in the Iranian uranium enrichment facility in Natanz, was the first attack against an Industrial Control System (ICS) that gained broad attention and raised security awareness in the area of Operation Technology (OT). Another attack targeting ICSs caused a power outage in Ukraine which affected 225.000 people [2]. More, often sophisticated, attacks have been carried out since [3].

Predominantly these attacks target PLCs. These differ in several important aspects from traditional computers: (i) They do not interact with data, but with the physical world instead. (ii) Thus, their main interest is not confidentiality, but reliably running a continuous process. (iii) Their lifespan is commonly between 15 and 20 years. (iv) Once installed, they hardly ever get patched. (v) They use proprietary firmware or operating systems (OS). (vi) They execute their programs in continuous, real-time cycles. Thus, due to their different primary objectives, design, and use they have to be treated differently [4].

In this work, we present an overview of known attack vectors against ICSs and show the underlying common security weaknesses (Section III). Secure coding practices and guidelines exist, but they are not the focus of this work. We touch on them only briefly in Section IV.

II. Taxonomy

There are many ways to structure ICS security [5][6][7]. We will use a target-based structure in our work for two reasons: (i) most publications presenting attacks follow this approach and (ii) due to proprietary protocols and operating systems these attacks are most often vendor specific. However, this approach can be seen as *pars pro toto* since many of the principles presented can be transferred from one vendor to another.

III. Attacks

In the past, an *air gap* (not connecting the OT and Information Technology (IT) networks) was perceived as adequate protection against attacks. However, with the advent of Manufacturing Execution Systems, remote access, and the Industrial Internet of Things this no longer holds. Next, Security by Obscurity (use of closed-source, vendor-specific protocols and security mechanisms) has numerously been compromised as we will show next.

All presented attacks assume that the adversary has already gained access to the IT network and can move laterally to the OT network, as has been the case in past attacks.

A. Siemens S7

After the Stuxnet attack in 2011, Beresford [8] was the first to exploit vulnerabilities of Siemens S7-300 and S7-1200. His work shows in detail how to gain access to a PLC by first capturing session data and then replaying it to the PLC. This can further be extended to altering the control logic or disabling the authentication mechanisms altogether without having access to the engineering workstation. These attacks are possible due to the use of insecure protocols in ICSs.

Following up on the replay attack, [9] extend this procedure by reverse-engineering the password encoding scheme. They succeed, revealing the custom eight-byte XOR encoding scheme. This allows not only to update the password of the PLC but also to clear arbitrary PLC memory which effectively renders the PLC useless.

As [10] remarks, the password can also be revealed by using an exhaustive search due to the small key space of only eight bits.

By implementing a PLC worm [11], PLCs can infect one another, having the worm automatically propagate through the whole OT network segment.

However, this worm can be detected since TIA Portal engineering software can be used to retrieve the code from the PLC. Building upon existing reverse engineering findings, [12] show that it is possible to disguise the code change. The

 ISBN: 978-1-68558-054-4

source code in the PLC exists in a *source object*, but when communicating with the PLC a *run object* is sent. This can be modified to a custom behavior, resulting in a different program being run in the PLC than shown on the engineering station.

B. Allen Bradley

Attacks against PLCs require data transmission via the network. This makes Network Intrusion Detection Systems (NIDS), such as Anagram [13], a natural countermeasure. However, it is possible to develop stealthy attacks by either modifying the signature of the packet header (*Data Execution Attack*) or by fragmentation of the data with added noise padding (*Fragmentation and Noise Padding*). Both attacks are successfully carried out in [14] without being detected by a NIDS and attacked a Schneider Modicon M221, as well as an Allen Bradley MicroLogix 1400.

Vendor-specific engineering software is used to send and retrieve compiled logic to and from the PLC. Thus, it can also be used for forensics in case of control logic injection attacks. However, as [15] show, this is no longer the case if a *Denial of Engineering Operations* attack is used. They show three different versions: (i) Hiding infected ladder logic from the engineering software, (ii) crashing the engineering software upon retrieving code from the PLC, and (iii) injecting a crafted ladder logic program to the PLC that crashes the engineering software. While the former two are *man-in-the-middle-attacks*, the latter is the stealthiest since it allows the attacker to leave the network after the attack. To detect these attacks, the authors also developed a decompiler for ladder logic that can completely restore the ladder program from network traffic and thus makes it possible to detect the injected control logic.

C. Schneider

The Schneider Tricon PLC employs Triple Modular Redundancy, using *two-out-of-three voting*. It is widely used in nuclear power plants. The software is downloaded simultaneously to all three processors, making this PLC susceptible to common mode failures induced by software, such as a cyber attack. Two attacks are proposed by [16], namely *latent attack* which downloads valid but incorrect control logic to the PLC, and *immediate failure attack* which transfers invalid data to the PLC, leading to a denial of engineering and an error on the PLC. While the first causes an incorrect behavior of the PLC and at the same time deceives the operator, the latter leads to a major downtime of the whole system since a complete reset and new program download becomes necessary.

A sophisticated attack against a Schneider Modicon M221 is shown in [17]. The authors propose the fully automated attack tool *CLIK* that consists of four stages: (i) stealing control logic binary from the PLC, (ii) decompiling the stolen binary to source code, (iii) infecting the control logic in the PLC, and (iv) concealment of infection from engineering software using a virtual PLC.

D. Beckhoff

With the use of common operating systems such as Windows CE or Windows 10, Beckhoff differs from other vendors who all use proprietary OS. However, this also makes the PLC susceptible to attacks known from the IT world. Bonney et al. [18] examine a CX5020 PLC and find several possible attack vectors due to plaintext transmitted connection setups (including user name and password), by default enabled webserver, and insecure default user name and password for Virtual Private Network.

E. Modbus

Modbus is a popular, vendor-agnostic protocol used in ICSs. Attacks against SCADA systems can be carried out using this protocol, as [19] show. The authors identify four attack classes: reconnaissance, response and measurement injection, command injection, and denial of service. For each class, they present several concrete attacks.

F. Open Platform Communications Unified Architecture

Open Platform Communications Unified Architecture (OPC UA) is a platform-independent service-oriented architecture that is widely used in the industry and supported by all major vendors. It is mainly used for non-real-time data exchange between PLCs and a variety of clients but with OPC UA over Time-Sensitive-Networking (TSN), it can also be used for real-time communication. OPC UA libraries exist for all major programming languages. OPC UA has been designed with a strong focus on security by integrating the following mechanisms: user security by using a user security token, application security by using digitally signed X.509 certificates, and transport-level security by signing and encrypting each message [20]. A report of the German Federal Office for Information Security (BSI) attests adequate protection against numerous threats, while *denial of service* and *server profiling* can only be reduced by its protection mechanisms [21]. It also notes that no systematic errors could be detected. However, as [22] note, provisioning has not been examined. They show that trust on first use (TOFU) is used for provisioning, thus undermining the security guarantees of OPC UA if the adversary gains access during this first phase. They also note that provisioning is both overly complex and often vaguely documented, leading to misconfiguration or disabling security features.

OPC UA security can also be weakened by major security flaws in its artifacts, as [23] show. The authors examine 48 OPC UA-enabled artifacts, both products from vendors and open-source libraries. Their main findings include disabled security(14.6%) and errors in trust list management (64.6%).

IV. Guidelines

To help secure ICSs, several guidelines have been published. The *National Institute of Standards and Technologies* (NIST) issued the comprehensive *Guide to Industrial Control Systems Security* [24] that provides a global picture of ICS security

 ISBN: 978-1-68558-054-4

both in technical and organizational terms. A similar guideline is available from the German BSI [25].

In analogy to the IT world, the US *Computer Emergency Response Team* (CERT) publishes alerts and advisories concerning ICSs [26]. In line with this practice, manufacturers of ICSs have started to publish advisories and security bulletins [27][28][29].

From a technical perspective, secure coding practices for ICSs are emerging and collected in an open-source effort [30].

V. Conclusion and Future Work

The notion of PLCs being insecure by design is a recurrent theme in all presented work, the weakest links being a lack of authentication mechanisms and insecure protocols. OPC UA, when properly implemented and set up is the exception to the rule. Mechanisms like Intrusion Detection Systems [31] can help harden industrial systems. However, this is only reasonable after basic security mechanisms like authentication and secure protocols are put in place. Forensics [32][33] provides the cornerstone for not experiencing the same attack several times and helps to build vulnerability databases. Secure coding practices promote a *defense-in-depth* approach and help to reduce attack surfaces once the adversary gained access. However, they are only an additional layer of protection and cannot compensate for the aforementioned weaknesses.

As of today, many of the PLCs in the field are not or are insufficiently protected. Future work will thus be twofold: Targeting existing devices that have many of the vulnerabilities presented here and finding means to mitigate these in newer devices. The latter will mainly need to find fast and at the same time efficient cryptographic algorithms. While this is achievable with hardware-based symmetric encryption as [34] show, effective software-based solutions are still to be researched. For the first, however, our recommendation is to find and standardize ways of penetration testing in ICSs, such as Metasploit [35] from the IT world. This can then also be used to automatically check assets against newly discovered vulnerabilities. However, due to the variety of vendor-specific protocols and operating systems, this will be a demanding task.

References

[1] R. Langner, "Stuxnet: Dissecting a Cyberwarfare Weapon," *IEEE Security Privacy*, vol. 9, no. 3, pp. 49–51, 2011.

[2] R. M. Lee, M. J. Assante, and T. Conway, "Analysis of the Cyber Attack on the Ukrainian Power Grid," SANS Industrial Control Systems, Tech. Rep., 2016.

[3] K. Hemsley and R. Fisher, "A History of Cyber Incidents and Threats Involving Industrial Control Systems," in *Critical Infrastructure Protection XII*. Cham: Springer International Publishing, 2018, pp. 215–242.

[4] O. El Idrissi, A. Mezrioui, and A. Belmekki, "Inadequacy of IT Approaches to Manage Cyber Security in ICS Context," in *Proceedings of the 12th International Conference on Soft Computing and Pattern Recognition (SoCPaR 2020)*. Cham: Springer International Publishing, 2021, pp. 678–689.

[5] T. Fleury, H. Khurana, and V. Welch, "Towards a taxonomy of attacks against energy control systems," in *International Conference on Critical Infrastructure Protection*. Springer, 2008, pp. 71–85.

[6] A. Flowers, S. Smith, and A. Oltramari, *Security Taxonomies of Industrial Control Systems*, 08 2016, pp. 111–132.

[7] Z. Drias, A. Serhrouchni, and O. Vogel, "Taxonomy of attacks on industrial control protocols," in *2015 International Conference on Protocol Engineering (ICPE) and International Conference on New Technologies of Distributed Systems (NTDS)*, 2015, pp. 1–6.

[8] D. Beresford, "Exploiting Siemens Simatic S7 PLCs," *Black Hat USA*, vol. 16, no. 2, pp. 723–733, 2011.

[9] H. Wardak, S. Zhioua, and A. Almulhem, "PLC access control: a security analysis," in *2016 World Congress on Industrial Control Systems Security (WCICSS)*, Dec 2016, pp. 1–6.

[10] A. Ayub, H. Yoo, and I. Ahmed, "Empirical Study of PLC Authentication Protocols in Industrial Control Systems," in *Fifteenth IEEE Workshop on Offensive Technologies (WOOT)*, 2021.

[11] R. Spenneberg, M. Brüggemann, and H. Schwartke, "Plcblaster: A worm living solely in the plc," *Black Hat Asia*, vol. 16, pp. 1–16, 2016.

[12] E. Biham, S. Bitan, A. Carmel, A. Dankner, U. Malin, and A. Wool, "Rogue7: Rogue engineering-station attacks on S7 Simatic PLCs," *Black Hat USA*, 2019.

[13] K. Wang, J. J. Parekh, and S. J. Stolfo, "Anagram: A content anomaly detector resistant to mimicry attack," in *International workshop on recent advances in intrusion detection*. Springer, 2006, pp. 226–248.

[14] H. Yoo and I. Ahmed, "Control logic injection attacks on industrial control systems," in *IFIP International Conference on ICT Systems Security and Privacy Protection*. Springer, 2019, pp. 33–48.

[15] S. Senthivel, S. Dhungana, H. Yoo, I. Ahmed, and V. Roussev, "Denial of engineering operations attacks in industrial control systems," in *Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy*, 2018, pp. 319–329.

[16] B. Lim, D. Chen, Y. An, Z. Kalbarczyk, and R. Iyer, "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)*, Jan 2017, pp. 205–210.

[17] S. Kalle, N. Ameen, H. Yoo, and I. Ahmed, "CLIK on PLCs! Attacking Control Logic with Decompilation and Virtual PLC," *Proceedings 2019 Workshop on Binary Analysis Research*, 2019.

[18] G. Bonney, H. Höfken, B. Paffen, and M. Schuba, "ICS/SCADA security analysis of a Beckhoff CX5020 PLC," in *2015 International Conference on Information*

 ISBN: 978-1-68558-054-4

Systems Security and Privacy (ICISSP), Feb 2015, pp. 1–6.

[19] T. H. Morris and W. Gao, "Industrial control system cyber attacks," in *1st International Symposium for ICS & SCADA Cyber Security Research 2013 (ICS-CSR 2013) 1*, 2013, pp. 22–29.

[20] OPC Foundation, "Practical Security Recommendations for building OPC UA Applications," 2018.

[21] Federal Office for Information Security, "OPC UA Security Analysis," 2016.

[22] F. Kohnhäuser, D. Meier, F. Patzer, and S. Finster, "On the Security of IIoT Deployments: An Investigation of Secure Provisioning Solutions for OPC UA," *IEEE Access*, vol. 9, pp. 99 299–99 311, 2021.

[23] A. Erba, A. Müller, and N. O. Tippenhauer, "Security Analysis of Vendor Implementations of the OPC UA Protocol for Industrial Control Systems," Apr. 2021.

[24] K. A. Stouffer, J. A. Falco, and K. A. Scarfone, "Guide to Industrial Control Systems (ICS) Security," 2011.

[25] BSI, "ICS Security Compendium," 2013, accessed: 2021-12-07. [Online]. Available: https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/ICS/ICS-Security_compendium.html

[26] CISA, "Industrial Control Systems," accessed: 2021-12-07. [Online]. Available: https://us-cert.cisa.gov/ics

[27] Siemens, "Siemens Security News," 2023, accessed: 2023-02-01. [Online]. Available: https://new.siemens.com/global/en/products/services/cert/news.html#/posts

[28] Beckhoff, "IPC - Security Guideline - Advisories," 2023, accessed: 2023-02-01. [Online]. Available: https://infosys.beckhoff.com/english.php?content=../content/1033/ipc_security/976057355.html&id=

[29] Schneider, "Cybersecurity support portal," 2023, accessed: 2023-02-01. [Online]. Available: https://www.se.com/ww/en/work/support/cybersecurity/security-notifications.jsp

[30] PLC Security, "Top 20 PLC Secure Coding Practices," accessed: 2021-12-07. [Online]. Available: https://www.plc-security.com

[31] D. A. Yeager, L. Vega, A. Brown, H. Burke, J. Petersen, E. Rodas, J. Welsh, A. Wen, and M. Zaron, "Case Study: Architecting a Solution to Detect Industrial Control System Attacks," in *2020 IEEE Systems Security Symposium (SSS)*, 2020, pp. 1–8.

[32] I. Ahmed, S. Obermeier, S. Sudhakaran, and V. Roussev, "Programmable Logic Controller Forensics," *IEEE Security Privacy*, vol. 15, no. 6, pp. 18–24, November 2017.

[33] M. Cook, I. Stavrou, S. Dimmock, and C. Johnson, "Introducing a forensics data type taxonomy of acquirable artefacts from programmable logic controllers," in *2020 International Conference on Cyber Security and Protection of Digital Services (Cyber Security)*, June 2020, pp. 1–8.

[34] M. Skuballa, A. Walz, H. Bühler, and A. Sikora, "Cryptographic Protection of Cyclic Real-Time Communication in Ethernet-Based Fieldbuses: How Much Hardware is Required?" in *26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2021, pp. 1–7.

[35] Rapid7, "Metasploit," 2023, accessed: 2023-02-01. [Online]. Available: https://www.metasploit.com