Secure Cyber-Physical Production Systems: Solid Steps towards Realization

Johanna Ullrich*, Artemios G. Voyiatzis*, Edgar R. Weippl*
* SBA Research, Vienna, Austria
(firstletterfirstname)(lastname)@sba-research.org

***Abstract*—Sophisticated production systems include plenty of information technology (IT) in order to gain more efficiency. However, this on-going development bears the drawback of lacking security. Cyber-physical production systems (CPPS) are likely to be struck by a cyber-launched attack; but might also be themselves the origin of an attack targeting IT infrastructures or other production systems. Far from an ideal solution, the involved engineering disciplines appear to work in parallel despite aiming for the same goal: securing the production systems.**

In this paper, we highlight small measures that are able to achieve large effects on CPPS security: (1) Extending interoperability testing by security testing gains robustness against intentionally malformed inputs; (2) the extension of today's models so that they enable the description of malicious actions would allow to assess system behavior in presence of an attack; and (3) a layered approach on CPPS security enables to address malicious activities at an adequate, semantic layer without the need for precarious shadow systems.

I. Introduction

The ever-increasing sophistication of production systems calls for the use of Information and Communication Technologies (ICT) components and the integration of deep supply chains comprising of different actors at each stage. The collective term "cyber-physical system" (CPS) is often used to emphasize the strong interlink and interdependence of the digital (cyber) and physical components of modern production systems consisting of complex interactions of systems of systems. The introduction of ICT components improves the production efficiency by collecting, analyzing, and acting upon information related to internal and external events. This includes, among other, the surrounding physical environment, the status of the machines (i.e., e-maintenance [1]), ample computing capability for more precise control, and timely demand response.

The advantages of ICT are profound. However, in a world of deep supply chains and complex operations, there is a rising concern regarding the security and trustworthiness of the ICT components [2], [3]. Can we trust the information collected from sensors? Can we trust that the actuators received the commands that were actually sent to them? Can we trust that the back-end algorithms are not manipulated? Are the control loops of the cyber-physical production system (CPPS) able to cope with intentionally-manipulated information? Are there new threats for the safety of the environment and humans, or for business continuity arising from the use of ICT?

Information Technology (IT) is a fast-evolving field and new vulnerabilities are constantly emerging. Currently, the most common approach to CPPS protection is to reuse existing IT security practices, such as access control, patching, firewalls and encryption. These mainly defend against already known attack vectors. The physical component of a cyber-physical system is not necessarily taken into account by these countermeasures, and so, in the absence of further protection, remains vulnerable. Even without this additional challenge, it can be difficult for system operators to keep up with innovations and hazards; given the complexity and size of cyber-physical systems, this security issue should be addressed urgently. In addition, the possibility of attacks exploiting the dynamic of a system's physical parts must be considered.

The role of security for realizing dependability, safety, and reliability is discussed in [4]. It is no longer feasible to use separate approaches to design and develop safe and secure CPS; the need to co-engineer CPSs using a combined safety and security development lifecycle is discussed in [5].

In this paper, we analyze how security can be addressed in a CPPS context towards bridging the gap between the fields of industrial and computer engineering. We provide indications that the two fields are actually different viewpoints of the same underlying problem. As such, we argue that the two fields are actually working in *parallel*, developing alike solutions albeit with different terminology and thus, there is a waste of resources and a danger to diverge and isolate. By means of three examples, we argue how CPPS interoperability, system design, and self-protection can actually benefit from common development.

II. Background

Security threats using ICT against physical (critical) infrastructures, such as water treatment facilities and power production plants are not new; reported incidents date back as far as the 1980s [6]. Attacks on such cyber-physical systems became more frequent since the early 2000s. Common targets include transport systems, power generation, and utilities. The metal working industry was also attacked: a steel mill was compromised in 2014, when attackers gained access to the relevant networks by means of spearphishing, and ultimately sabotaged physical components of the plant [7]. In the same year, numerous energy companies became victims of a hacking group known as "Dragonfly". Although the attack methods were similar to the previous ones (e-mail attacks, malware), cyber-espionage seems to have been the main goal of Dragonfly [8]. In the case of cyber-physical production systems,

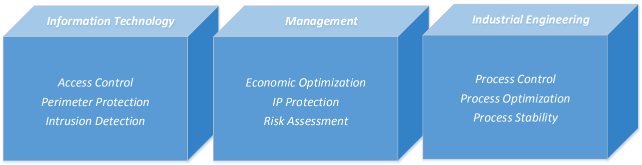

Fig. 1. Security goals of different CPPS stakeholders

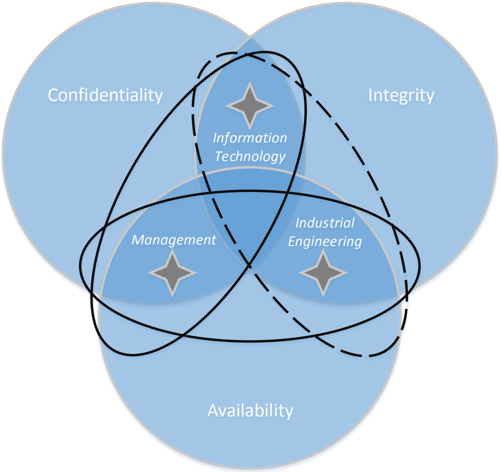

Fig. 2. Security goals of CPPS stakeholders according to the AIC/CIA triad

it is just a small step from data theft to damaging physical components or whole (critical) infrastructures. Therefore, every CPPS must be protected as comprehensively as possible.

There are multiple stakeholders involved in CPPS operation. The notion of security for each of them can be quite different. We can group the internal stakeholders of a CPPS under three classes, as depicted in Figure 1. The information technology class is concerned about ensuring that the information handled by the CPSS can be trusted. As such, it is acceptable to reduce CPPS availability in order to ensure information confidentiality and integrity. The production engineering class is concerned about retaining the control of the (process or part) production. This includes ensuring process stability and optimizing the resource consumption (e.g., manpower, time, and energy). As such, it is acceptable to reduce confidentiality and integrity in favor of process longevity. Finally, the management class can be considered as a middle point between these two; it is concerned about business continuity in the long term. As such, risk assessment can define preference of information confidentiality (e.g., intellectual property and business data protection) over short-term economic benefit (e.g., retaining specific production levels).

In the past, the triad of confidentiality, integrity, and availability (CIA) has been the common ground for the management and information technology stakeholders. The reverse triad (AIC) has been the common ground for management and production engineering stakeholders. However, in the case of CPPS there are no more boundaries between the two (cyber and physical) infrastructures and control. All three stakeholders must reach consensus albeit aiming for different notions of security, as depicted in Figure 2.

III. Integrating Security into CPPS Engineering

We discuss in the following how the security engineering practice can be integrated seamlessly into the CPPS design and development practice using three example cases, namely interoperability, system design, and self-protection. This common development can benefit the overall security and safety of a CPPS.

A. *Interoperability*

CPPS comprise heterogeneous technological components which are not necessarily compatible with each other. Standardization of data models and protocols for information exchange allow to address this complexity. Suitable middleware acts as an intermediary, parsing data streams under specific assumptions for their structured format, and translating to representations that can be locally processed. There is a direct analogy with Internet servers accepting data streams claiming to comply with a network protocol (e.g., an HTTP request). Such software has been proven to be very hard to implement correctly and is often very fragile when exposed to maliciously-crafted input. Various software testing approaches have been used for black-box or white-box security testing, including input fuzzing [9], model-based checking, and combinatorial testing [10].

Testing for interoperability is widely practiced in industrial systems with the aim to ensure that different middleware implementations can exchange information with each other. However, security testing is in most times out of consideration, as the main focus is the compliance with a specification and correct mapping of the semantics. Furthermore, the threat models for industrial systems often do not account for malicious actions as the originators of these actions are considered to be isolated from the system. This has changed with remote maintenance, and lately the introduction of consumer-grade devices at the factory level, often administered by the personnel itself (BYOD, bring-your-own-device), constitutes an even more powerful attack surface that cannot be ignored anymore.

In this setting, we argue that a closer collaboration of the two "worlds" is beneficial for both of them. From the security point of view, it is resource- and time-consuming to setup the necessary expert teams, equipment, platforms, and workflows for performing just a few experiments. From the interoperability point of view, all these are already in place and the integration of security testing requires minimal additional effort. However, there is a significant benefit in this case: the middleware will not only be tested for handling correctly the "correct" inputs or the occasionally "erroneous" input (as testing for safety would imply). Rather, it will be tested for robustness against input that is maliciously crafted on purpose. This can increase significantly the trustworthiness of the middleware and the confidence level that it will react in a secure and safe way after many years of operation.

B. *System design*

The design of a CPPS is a complex procedure by itself. The ability to replicate its behavior is rather limited as emulation and simulation at the scale of a full CPPS are not a realistic option. Even at a component level, it is not always feasible to predict the behavior of the system as the available models are developed with strict assumptions. This is necessary in many cases to cope with the wide complexity of the involved physical processes. Robust control algorithms sufficiently defend against *accidental* and *random* events but they are not able to cope with sustained *malicious* events that are carefully crafted to occur at specific times. Falling to safe operation conditions by reducing the production capacity is always an option. However, if the problem originates from malicious actions, then it can be used consistently exploited so as, from a security point of view, to realize a denial-of-service attack against the production system by guiding it to operate in reduced capacity for safety reasons.

The need to adapt the design practices towards co-engineering for safety and security is already discussed and early indications from case studies are also available [5]. The complexity of systems of systems drives the need for new and improved testing approaches at a system rather than a component level. There is a pressing need to improve the availability of testbeds (emulation or simulation) and the capability to integrate more realistic conditions in the simulation environments. As such, it is necessary to evolve existing platforms by interconnecting them or developing missing functionality (e.g., Matlab/Simulink models of physical processes that interact with network traffic simulators for Internet node behavior). Last but not least, it is necessary to extend the models for control algorithms so that they can describe and integrate malicious actions [11]. The coordinated development in the aforementioned topics can be beneficial for both industrial and (information) security engineering, as it will allow to increase the degree of realism and provide better insights for the design and the future revisions of a CPPS.

C. Self-protection

The realization of self-protection functionality inside a CPPS must be a consideration as early as the design phase of the system. There are already numerous attempts to address the issue of security at different layers of abstraction. However, most of these proposals lack a holistic approach in the design of security solutions. Rather, they aim to address security at a selected layer. This is not an optimal strategy for at least two reasons. The first is that the domain knowledge to address a security problem might not be available at the selected layer. For example, the requirement to address network-level denial-of-service attacks at the control layer or to identify fraudulent sensor readings from remote locations within the control loop are not realistic. Rather, the network layer should be able to cope with such malicious behaviors.

The second is that there is the danger to end up realizing "shadow" systems [12], which replicate the behavior of a component (e.g., a SCADA server) within a different layer (e.g., a network intrusion detection system). This is a profound waste of resources (more configuration, more maintenance) and an increase of the available attack surfaces against the CPPS. Even worse, one might end up with conflicting views of the same problem and face a conflict of decision: which of the two views should be considered as the authoritative one?

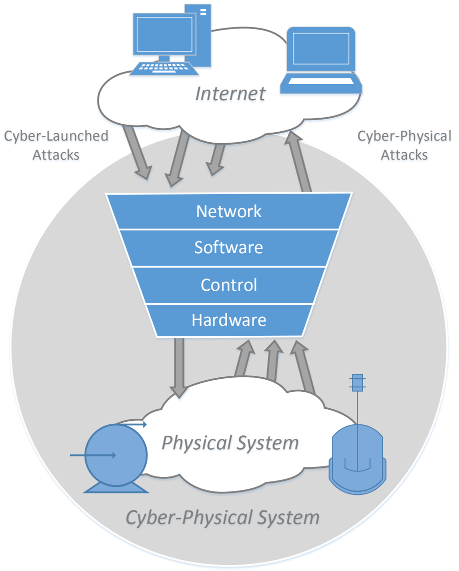

Fig. 3. Cyber-launched and cyber-physical attacks

We claim that, borrowing from secure network architecture design [13], a layered design approach for secure CPPS can be beneficial for all involved parties. We consider six layers of security, as depicted in Figure 3. We expect that each of the four major layers (network software, system software, control logic, and hardware) can only filter some of the attacks. The layer approach allows each lower level to focus on defending against attacks that carry the semantics of the specific layer. This way, no single layer is overwhelmed with security-related events and appropriate countermeasures can be designed with relaxed assumptions. For example, a CPPS component that implements in software a specific control algorithm will never be exposed to a storm of input values that might result in crash or extended periods of computation. Rather, the network layer will enforce that only a specific, sustainable rate of information packets will ever reach this component. We should also not overlook that the attacks do not originate from the Internet only. Rather, the Internet infrastructure itself is potentially on the receiving end of the attacks as well, as discussed in [14].

The security of ICT systems is most often engineered using a castle analogy: strategically-located, with a well-defended perimeter (walls and watchtowers), and controlled entry points. The static nature of ICT systems gives the attackers an extremely valuable and asymmetric advantage: *time*. In the case of a CPPS, this lifetime can span in decades that attackers can spend for monitoring a deployed system for detecting vulnerabilities and deciding the best time to launch an attack. When successful, attackers enjoy on average more than 200 days of free reign in breached environments – or even decades [15] – and affected organizations are notified mostly by outsiders (e.g., law enforcement agencies) about the breach [16].

The Moving Target Defense (MTD) strategy was proposed in the information systems security domain as a means to break this asymmetry. The idea of MTD is to explore system options so as to introduce a notion of "motion" and transform the system into a moving target for an attacker. One simple example is changing in a controlled manner the IP address of a computing system (attack surface) within an acceptable range of addresses (exploration surface). Provided that such changes are permitted and that the exploration surface is significantly greater than the attack surface, the workload for an attacker also increases. Such an option has been theoretically studied for IPv6-enabled nodes in wireless sensor networks used for home automation systems and smart meters [17]. We still lack real-world practical applications of MTD, although there are already attempts to systematize a theory of MTD [18]. We claim that already available architectures and technologies (e.g., cloud computing, network function virtualization (NFV), and software-defined networks (SDN) [19]) as well as defenses based on the IPv6 protocol (e.g., [20], [21]) can be used to realize MTD approaches. Given the time span of a CPPS, we argue that MTD is a viable alternative for breaking the asymmetry with respect to time and improving the security of a CPPS. In this sense, MTD can be one means for realizing self-protection mechanisms in a CPPS.

IV. Conclusions and future work

In this short paper, we discussed emerging issues relating to the security for cyber-physical production systems. We demonstrated three cases were industrial and computer engineering are actually working towards the same aims at the same time but from different perspectives. As such, there is potential for collaboration of mutual benefit rather than competition. We identified how this potential can be exploited for introducing security testing within interoperability testing, improving testbeds and used models to account and experiment on malicious actions, and implementing layered and moving-target defenses as an integral part of a CPPS towards achieving self-protection.

We share the view that the technological solutions are readily available but there is a lack of common terminology and mutual understanding. Thus, we aim to explore further the development of a consistent notation for the two fields, including mathematical, textual, and graphical explanations. Furthermore, there is a need to update the study curricula and provide more interdisciplinary education to the next generation of engineers. It is a long way but the first attempts, such as the one described in [22], demonstrate that it is indeed feasible.

Acknowledgment

This research was supported by the Bridge Early Stage grant P842485 and the COMET K1 program of the Austrian Research Promotion Agency (FFG).

References

[1] E. Jantunen, C. Emmanouilidis, A. Arnaiz, and E. Gilabert, "Economical and technological prospects for e-maintenance," *International Journal of System Assurance Engineering and Management*, vol. 1, no. 3, pp. 201–209, 2011.

[2] J. Boyens, C. Paulsen, R. Moorthy, N. Bartol, and S. A. Shankles, "Supply chain risk management practices for federal information systems and organizations," *NIST Special Publication*, vol. 800, no. 161, p. 1, 2014.

[3] R. George, "Why we should worry about the supply chain," *International Jornal of Critical Infrastructure Protection*, vol. 11, pp. 22–23, 2015.

[4] D. Serpanos and A. Voyiatzis, "Security challenges in embedded systems," *ACM Transactions on Embedded Systems*, vol. 12, no. 1s, pp. 66:1–66:10, March 2013.

[5] C. Schmittner, Z. Ma, and E. Schoitsch, "Combined safety and security development lifecylce," in *Industrial Informatics (INDIN), 2015 IEEE 13th International Conference on*, July 2015, pp. 1408–1415.

[6] "RISI: The repository of industrial security incidents." [Online]. Available: http://www.risidata.com/Database/event_date/desc

[7] R. Lee, M. Assante, and T. Conway, "SANS ICS Defense Use Case (DUC) Dec 30, 2014: ICS CP/PE case study paper - German Steel Mill Cyber Attack," 2014. [Online]. Available: https://ics.sans.org/media/ICS-CPPE-case-Study-2-German-Steelworks_Facility.pdf

[8] J. Langill, E. Zambon, and D. Trivellato, "Cyberespionage campaign hits energy companies," 2014. [Online]. Available: http://www.secmatters.com/sites/www.secmatters.com/files/documents/whitepaper_havex_US.pdf

[9] A. Walz and A. Sikora, "Testing embedded TLS implementations using fuzzing techniques and differential testing," *BW-CAR SINCOM*, p. 36, 2015.

[10] J. Bozic, B. Garn, I. Kapsalis, D. Simos, S. Winkler, and F. Wotawa, "Attack pattern-based combinatorial testing with constraints for web security testing," in *Software Quality, Reliability and Security (QRS), 2015 IEEE International Conference on*, Aug 2015, pp. 207–212.

[11] A. A. Cardenas, S. Amin, and S. Sastry, "Secure control: Towards survivable cyber-physical systems," in *International Conference on Distributed Computing Systems Workshops*. IEEE, 2008, pp. 495–500.

[12] M. C. Dacier, F. Kargl, v. d. R. Heijden, H. König, and A. Valdes, "Network attack detection and defense: securing industrial control systems for critical infrastructures," *Informatik Spektrum*, vol. 37, pp. 605–607, 2014.

[13] D. Serpanos and A. Voyiatzis, "Secure network design: A layered approach," in *2nd International Workshop on Autonomous Decentralized System (IWADS 2002)*, 2002, beijing, P.R.China, November 6-7, 2002.

[14] Y. Oren and A. D. Keromytis, "From the Aether to the Ethernet—attacking the Internet using broadcast digital television," in *23rd USENIX Security Symposium (USENIX Security 14)*. San Diego, CA: USENIX Association, Aug. 2014, pp. 353–368.

[15] D. Goodin, "How "omnipotent" hackers tied to NSA hid for 14 yearsand were found at last," 2015. [Online]. Available: http://arstechnica.com/security/2015/02/how-omnipotent-hackers-tied-to-the-nsa-hid-for-14-years-and-were-found-at-last/

[16] M. Sherburne, R. Marchany, and J. Tront, "M-trends 2015: A view from the front lines," 2015.

[17] ——, "Implementing moving target IPv6 defense to secure 6LoWPAN in the Internet of Things and smart grid," in *Proceedings of the 9th Annual Cyber and Information Security Research Conference*, ser. CISR '14. New York, NY, USA: ACM, 2014, pp. 37–40.

[18] R. Zhuang, S. A. DeLoach, and X. Ou, "Towards a theory of moving target defense," in *Proceedings of the First ACM Workshop on Moving Target Defense*. ACM, 2014, pp. 31–40.

[19] I. Ahmad, S. Namal, M. Ylianttila, and A. Gurtov, "Security in software defined networks: A survey," *Communications Surveys Tutorials, IEEE*, vol. 17, no. 4, pp. 2317–2346, 2015.

[20] J. Ullrich, K. Krombholz, H. Hobel, A. Dabrowski, and E. Weippl, "IPv6 security: Attacks and countermeasures in a nutshell," in *USENIX Workshop on Offensive Technologies (WOOT)*, 2014.

[21] J. Ullrich and E. Weippl, "Privacy is not an option: Attacking the IPv6 privacy extension," in *Proceedings of the International Symposium on Research in Attacks, Intrusions, and Defenses (RAID)*, 2015, pp. 448–468.

[22] T. Nguyen, M. Gondree, and D. Reed, "Teaching industrial control system security using collaborative projects," in *Proceedings of Conference on Cybersecurity of Industrial Control Systems (CyberICS)*, 2015.