Behavioral Intrusion Detection Methodology for Hybrid Industrial Control Systems

Estelle Hotellier, Franck Sicard, Julien Francq, Stéphane Mocanu

Behavioral Intrusion Detection Methodology for Hybrid Industrial Control Systems

Estelle Hotellier*†, Franck Sicard*, Julien Francq* and Stéphane Mocanu†
*Naval Cyber Laboratory, Naval Group
83190 Ollioules, France
{estelle.hotellier, franck.sicard, julien.francq}@naval-group.com
†Laboratoire d'Informatique de Grenoble
Univ. Grenoble Alpes, CNRS, Inria, Grenoble INP, 38000 Grenoble, France
{estelle.hotellier, stephane.mocanu}@inria.fr

***Abstract*—This paper presents the first results of a Ph.D started in April 2021. The topic is the intrusion detection in complex Industrial Control Systems (ICSs). We are interested in Process-Aware attacks i.e. attacks that target the physical integrity of systems. We consider the hybrid nature of ICSs and our methodology applies for event-driven and continuous dynamical systems. We aim at developing a behavioral network traffic Intrusion Detection System (IDS) based on the ICS characterization through security properties. To do so, we extract system safety properties from standards, devices programs or system specifications and synthesize them into security patterns. These patterns are then monitored by our IDS which is in charge of raising alerts.**

I. Introduction

Industrial Control Systems (ICSs) are networks of physical and computational assets, interconnected together for the execution of tasks in an industrial environment [1]. They can be found in various sectors such as transportation, energy supply, manufacturing, defense, etc. Such systems are vulnerable and the number of malicious actions targeting ICSs is constantly rising since 2010 [2]. 2010 is the year when the well-known Stuxnet cyberattack occurred which is one of the first ICS tailored attack (Natanz Nuclear Facility, Iran) [3]. In 2016, CrashOverride attack successfully caused a power cut in Ukraine for a few hours [4]. In 2021, a Florida water treatment facility was victim of an attack aiming to modify chemical levels in the city water supply [5].

These attack examples emphasize the need for security in industrial systems. Originally, ICSs were operating as isolated networks and they had to face accidental risks only. But last decades advancements brought new technologies and an increasing interconnectivity of assets. As a result, industrial systems now have to deal with external risks and more specifically network attacks.

Intrusion Detection Systems (IDSs) follow two categories: signature-based and behavior-based [6]. The first category relies on specific recognition of malicious behaviors and is able to detect already known attacks. The second category relies on the definition of a normal behavior for a system and leverage the detection of unknown attacks. On our side, we focus on behavioral IDSs based on network traffic.

In the next Section, we present the considered system characteristics. Section III describes the threat model and the related attack typology. In Section IV, we detail our attack detection approach. We then discuss the obtained results in Section VI. Section VII is a state of the art. We conclude and present future work in Section VIII.

II. Industrial Control Systems Characteristics

In our approach, we consider complex and hybrid ICSs, i.e. with hierarchical and distributed control. It means that several control objectives coexist in order to enforce a global correct behavior of the system. The global system is composed of sub-processes, locally controlled. The typical architecture of an ICS [7] will show *local loops* which are the elementary building blocks of such complex systems (they are generally composed of a local controller, sensors and actuators). These local loops enforce lower level control objectives such as motor speed or position for instance. Higher level objectives like trajectory tracking are ensured by higher level controllers that take care of synchronizing local loops. These higher level controllers are usually Programmable Logic Controllers (PLCs). The highest level is the Supervisory Control And Data Acquisition (SCADA) that allows coordination between the lower levels and a view of the process data for the human operators. It is important to state that the closest the devices are to the physical process, the more real-time performances are expected. On the contrary, at the PLC level, soft-real time is required and no real time at the SCADA level.

We also consider ICS hybrid nature. Indeed, real-life ICSs have both continuous-time dynamics and discrete event-driven dynamics [8]. An event-driven dynamic is characterized by a change of state on the occurrence of an event such as the reaching of a position for example. Whereas for time-driven dynamics, the systems state is constantly evolving with time (e.g., the water filling of a tank).

III. Threat Model

We are interested in Process-Aware attacks. These attacks require a deep knowledge of the physical system and make use of process data such as process variables values, devices parameters, control logic, etc. Furthermore, these attacks use

legitimate frames that perfectly respect communication protocols. To give an example, let us consider a tank in which two liquids A and B are to be mixed together respecting the following operating procedure: valve A is opened until level l_A is reached, after what valve B is opened until l_B. In that context, an attack could consist in inverting the order of opening for valve A and valve B, leading to a sabotage of product dosages. The main characteristic of such an attack is the fact that only legitimate commands were sent. The protocol specifications and frame characteristics were perfectly respected. Only the order in the sequence of commands was affected.

IV. Intrusion Detection Approach

Our contribution is a behavioral network IDS. The distinctive feature of our approach is to use security properties to characterize the normal behavior of the system. The first step is to construct the model which will be detailed in the following subsection. We will then describe its functioning at runtime.

A. Model construction

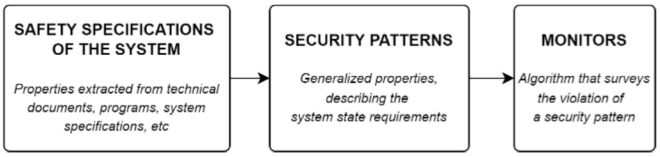

Fig. 1. Construction of the monitoring model

Fig. 1 represents the steps for the construction of our detection model. The aim of our approach is to link safety properties and security properties. According to IEC 62443 standard, *safety* is "to be free of unacceptable risk" [9] whereas *security* considers potential threats due to malevolent actions [10]. The security patterns used in our contribution are deduced from the safety specifications of the physical process. Safety specifications can be extracted from standards, technical documents of devices, controllers programs and so on. These safety properties are then expressed through generalized temporal patterns that we call the security patterns. There are several classes of temporal patterns in the literature. For our security patterns, we use qualitative patterns from the studies by Dwyer *et al.* [11], quantitative patterns from Konrad *et al.* [12] and also an extension of the quantitative patterns for continuous signals from Maler *et al.* [13]. Therefore, our security patterns are used for both time-based and event-based dynamics. Finally, each security pattern is associated to a monitor which is an algorithm that evaluates the violation of a pattern.

To give an example, consider again our tank system example. A safety specification can concern the order of the valve opening. These openings are events and the property can be synthesized by a precedence pattern from [11] as: *Precedence* (event $valve\ A \uparrow$ must occur before event $valve\ B \uparrow$).

B. Runtime Monitoring

Fig. 2 shows the structure of the runtime monitoring. At runtime, during the system functioning phase, each monitor surveys whether the pattern formula is satisfied or not based on the analysis of network traffic. A violation of a security pattern leads to the raise of an alert.

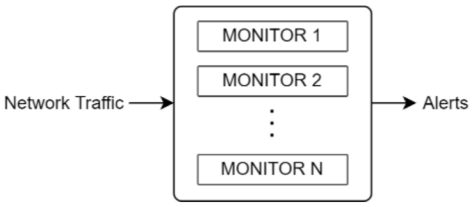

Fig. 2. Runtime monitoring

V. Implementation

We implemented our approach on a real ICS: a commercial educational workbench from Schneider Electric. The system is composed of a Human Machine Interface (HMI) as the supervisory control, a PLC and two local loops. Each loop comprises an industrial servo drive, an actuator (motor) and a sensor (encoder). It is a turnkey system with control programs provided by Schneider, therefore the codes are certified to be conform with industrial standards. The communication between the HMI and the PLC is made through a Modbus TCP/IP network and the communication between the PLC and local loops uses a CANopen fieldbus.

For our IDS system, we rely on the open-source Network IDS Zeek v4.1.0-dev.704[1]. Zeek is highly configurable and offers the possibility to implement sophisticated rules in its own scripting language. Furthermore, it is able to raise not only alerts but also general events. These two characteristics allow us on the one hand to extend Zeek framework to fieldbuses (Zeek beeing by default limited to Ethernet-based protocols) and on the other hand to easily implement our security pattern monitors.

The architecture of our contribution is illustrated in Fig. 3. The Workers are software agents which intercept the frames and filter network traffic. They transmit frames to the server by publishing messages. The server is a broker agent which centralizes and dispatches the messages. Detection scripts contain the monitors and they subscribe to certain types of messages and log alerts.

In our implementation, we only use the CAN worker yet, since our first results concern CANopen local loops only. We developed a Zeek CAN module based on the SocketCAN[2] kernel module for frame capture. In Fig. 3, Workers for other protocols are represented since our approach aims to be distributed and extended to other protocols in future works (we already developed the Modbus RTU Worker).

[1]https://www.zeek.org
[2]https://www.kernel.org/doc/html/latest/networking/can.html

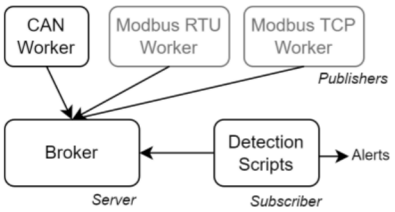

Fig. 3. IDS implementation architecture

VI. Evaluation

Our implementation include 33 security pattern monitors. We deployed attack scripts that aims to bring the system in an unsafe state such as interrupt operations, change order in sequence of events, force out of range values for process data, etc. We successfully detected all the attacks, since they were specifically designed to violate our security properties. There are no false negatives but some alerts overlap. Indeed, a single attack can trigger multiple monitors.

Concerning time performance, we evaluated the execution time of the detection algorithm depending on the number of monitors. We ran tests for 400 executions with a number of monitors going from 5 to 500. The results showed that the complexity of our system is proportional to the number of local loops.

VII. Related Work

The majority of behavioral IDSs in the literature focus on network traffic analysis but without considering physical process data [14]–[16]. These approaches are most of the time inefficient against Process-Aware attacks.

Among studies that focus on Process-Aware attacks, in [17] the approach is similar to ours in the monitor construction methodology. It differs in the detection level, which is between the HMI and PLCs. However, the detection is limited to sequential programs and requires knowledge of PLCs programs. Another complementary approach is the one in [18]. The systems specifications are also used in order to extract security properties for discrete and continuous variables. The authors fully characterize the critical states and the current state of the system is evaluated through its distance to a critical state. The approach was further refined for discrete systems in [19]. We assert that our approach is less complex since we do not build the joint state space of the system.

VIII. Conclusion and Future work

We presented a behavioral network IDS for complex, hierarchical and hybrid ICSs. The system is deployed at the local loops level of the system and is based on the monitoring of security patterns inherited from the safety specifications of the system. Our approach successfully detects Process-Aware attacks and has a low complexity which validates the scalability of our IDS. From a practical side, the implementation is made with Zeek IDS and is extended for fieldbuses support, event analysis and monitoring.

We aim at deploying a distributed IDS allowing cross-network detection: we want to be able to handle multiple fildbuses protocols but also different network hierarchical levels. Moreover, in order to cope with the overlap in the alerts, we want to develop an alert correlation approach that would allow to better comprehend and enrich attack scenarios.

References

[1] K. Stouffer, V. Pillitteri, S. Lightman, and M. Abrams, "Guide to Industrial Control Systems (ICS) Security," NIST SP 800-82 Rev. 2, Tech. Rep., 2015.

[2] "Threat Landscape for Industrial Automation Systems in 2019," ICS-CERT, Tech. Rep., 2020.

[3] N. Falliere, L. O. Murchu, and E. Chien, "W32.stuxnet dossier," Tech. Rep., 2011.

[4] J. Slowik, "Crashoverride: Analysis of the threat to electric grid operations," Dragos inc., Tech. Rep., 2019.

[5] "Recommendations following the oldsmar water treatment facility cyber attack," Dragos inc., Tech. Rep., 2021.

[6] H. Debar, M. Dacier, and A. Wespi, "Towards a taxonomy of intrusion-detection systems," vol. 31. Computer Networks, 1999.

[7] "IEC 62264: Enterprise-control System Integration," Industrial Electrotechnical Commission, Technical Specification, 2010.

[8] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*, Springer, Ed., 2008.

[9] "IEC 62443: Industrial communication networks - Networks and system security - Part 1-1: Terminology, concepts and models," Industrial Electrotechnical Commission, Technical Specification, 2009.

[10] S. Kriaa, L. Pietre-Cambacedes, M. Bouissou, and Y. Halgand, "A survey of approaches combining safety and security for industrial control systems," in *Reliability Engineering & System Safety*, vol. 139, 2015.

[11] M. B. Dwyer, G. S. Avrunin, and J. C. Corbett, "Patterns in property specifications for finite-state verification," in *Proceedings of the international conference on Software engineering*. ICSE, 1999.

[12] S. Konrad and B. H. C. Cheng, "Real-time specification patterns," in *Proceedings of the International Conference on Software Engineering*. ACM, 2005.

[13] O. Maler and D. Nickovic, "Monitoring temporal properties of continuous signals," in *International Symposium on Formal Techniques in Real-Time and Fault-Tolerant Systems*. Springer, 2004.

[14] N. Goldenberg and A. Wool, "Accurate modeling of Modbus/TCP for intrusion detection in SCADA systems," in *International Journal of Critical Infrastructure Protection*, vol. 6, no. 2, 2013.

[15] M. Caselli, E. Zambon, and F. Kargl, "Sequence-aware Intrusion Detection in Industrial Control Systems," in *Proc. ACM Workshop CPSS*, 2015.

[16] B. Ferling, J. Chromik, M. Caselli, and A. Remke, "Intrusion detection for sequence-based attacks with reduced traffic models," in *Measurement, Modelling and Evaluation of Computing Systems*. Springer International Publishing, 2018.

[17] O. Koucham, S. Mocanu, G. Hiet, J.-M. Thiriet, and F. Majorczyk, "Efficient Mining of Temporal Safety Properties for Intrusion Detection in Industrial Control Systems," in *IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes*, 2018.

[18] I. Nai Fovino, A. Coletta, A. Carcano, and M. Masera, "Critical State-Based Filtering System for Securing SCADA Network Protocols," in *IEEE Transactions on Industrial Electronics*, 2012.

[19] F. Sicard, É. Zamaï, and J.-M. Flaus, "An approach based on behavioral models and critical states distance notion for improving cybersecurity of industrial control systems," in *Reliability Engineering and System Safety*, vol. 188. Elsevier, 2019.