Distance Concept Based Filter Approach for Detection of Cyberattacks on Industrial Control Systems

Franck Sicard, Eric Zamaï, Jean-Marie Flaus

Distance Concept Based Filter Approach for Detection of Cyberattacks on Industrial Control Systems

Franck SICARD*. Éric ZAMAI*. Jean-Marie FLAUS*

*Univ.Grenoble Alpes, CNRS, GSCOP, 38000 Grenoble, France
(e-mail: franck.sicard@grenoble-inp.fr, eric.zamai@grenoble-inp.fr, jean-marie.flaus@grenoble-inp.fr)

Abstract: Industrial Control Systems (ICS) have become a new target of attackers since the beginning of the century. Computer worm Stuxnet proved the vulnerability of these systems to cyber-attacks. Control-command architecture is built to ensure the safety and the reliability of the system and the environment, however, several attacks or studies have underlined the lack of protection of components in an ICS. They equally proved the incomplete solutions proposed by the Information technology (IT). This paper presents an innovative approach for intrusion detection system in ICS based on the notions of states and distance between sets of states. Distance assessment over time between common and forbidden states of the system provides the prediction and discrimination of deviations. A proposed algorithm analyses orders sent to actuators continuously and enables to stop dangerous orders for the system. This study is supported by simulations inspired by classical ICS.

Keywords: Fault detection, Diagnosis, Reliability, ICS Cyberattacks, Behavioral Model, Discrete Event System, Manufacturing process.

1. INTRODUCTION

Industrial Control Systems (ICS) are integrated in many areas and critical infrastructures such as energy production and distribution (electricity, water, oil and gas, …), manufacturing systems, transportation, health services or defense (Fourastier and Pietre-Cambacedes, 2015). These systems have been designed to solve production issues by insuring productivity and reliability. ICS, as represented on Fig. 1, are composed of several hierarchical layers. A Supervisory Control and Data Acquisition *(SCADA)* system (level 2), shows data to operators, gathers data from field layer and transmits control information to the whole architecture of control-command. A control layer (level 1) with Remote Terminal Units *(RTU)*, controls devices of the field layer, acquires data from the field and communicates with SCADA system. The main component of RTUs is the *Programmable Logic Controller (PLC)* which based on the programmed logic and data sent by physical layer, decides actions (from control model) that have to be applied on the process. A low-level layer (level 0) composed of *Sensors / Actuators* makes the link between physical and cyber layers and transforms an initial product to a final product. Industrial *Communication networks* connect the layers. Recent architectures use devices communicating mainly with TCP/IP protocol. In consequence, ICS have been weakened because safety is not considered.

Since the beginning of the century, ICS are targeted by hackers that exploit vulnerabilities towards cyber-attacks. (Fourastier and Pietre-Cambacedes, 2015; McLaughlin et al., 2016) present a complete history of attacks. The reference of these attacks on ICS is computer worm Stuxnet (Falliere et al., 2011). PLC was targeted through office network and control program was then replaced by others which destroy the process after a learning phase. The interest of hackers for industrial systems is the physical impact of a cyber-attack. Indeed, if an attack succeeds, important damages are inflicted to the process (production shutdown, long repair time) and the environment (impact on human, health, ecology, social, financial loss).

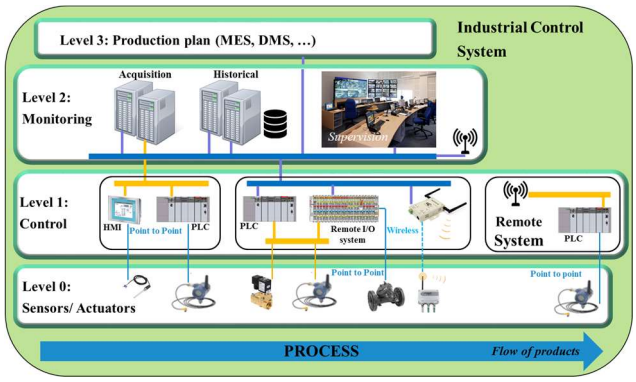

Fig. 1. Illustration of the architecture of an ICS

An exhaustive list of vulnerabilities is available on (ICS-CERT, 2016), different kind of attacks is described in (Fovino et al., 2012) as unauthorized command execution, Scada-Dos, Man-in-the-middle or Replay attack. All these attacks are inherited from the IT. Other attacks are inherent to level 1 and 0 of the ICS as *random attacks* where information is sent without taking into account the process, *sequential attacks* (Caselli et al., 2015; Li et al., 2016) when there is a violation of the sequential order of the control command and *false data injection* (Wang et al., 2014) where data is intercepted and modified between sensors and PLC or between actuators and PLC. Our study only focuses on attacks on the command between the PLC and actuators, because ICS architecture is vulnerable and it is the last occasion to stop any illicit order.

Approaches in the field of monitoring and especially in the one of detection can provide solutions to secure ICS from cyber-attacks. The aim of an attack is to have a service lost effect. As a result, security issues of ICS join the already established and classic problem of detection where different approaches are known to detect failures in a system. However, how can we differentiate between attacks (intentional) and failures (non-intentional)? The proposed approach is based on the concept of distance and trajectory, by designing patterns typical to an attack and the basis is the notion of distance from unsafe states, which over the time, gives information about hypothetical attacks. This article proposes to contribute to the notion of distance as a way to detect attacks.

2. TOWARDS DETECTING ICS ATTACKS USING DISTANCE CONCEPT

In this field of research, Intrusion Detection System (IDS) is generally proposed to detect attacks against computer systems and networks. (Denning, 1987) provides an IDS framework still used and efficient in Information Technology (IT). IDS are essentially a posteriori security measure where detection occurs after an intrusion. The basic idea is no matter the means used to ensure the security of a system, there will always be a residual risk that an intrusion occurs. The purpose of an IDS is to automatically identify violations of the system security policy. To do so, the IDS is based on data acquired by probes from the environment to protect. Different aspects of IDS as formalized by (Mitchell and Chen, 2014) have to be considered towards the choice of detection method and data source. Some IDS detect deviations from a behavior model (behavioral approach) and others rely on abnormal behavior knowledge database (signature approach). IDS that monitors network are called *Network Intrusion Detection System* (NIDS) and those using host data are called *Host Intrusion Detecting System* (HIDS).

Considering techniques that are investigated to protect ICS, IDS is a well-established field of research, and an exhaustive list of these techniques can be found in (Fourastier and Pietre-Cambacedes, 2015; McLaughlin et al., 2016). The signature approach, based on recognition of specific behavior as in (Pan et al., 2013), has some problems. The main issue of the approach is the non-detection of *zero-day* attack. However, ICS have the particularity to control physical process, so that behavioral approaches, or model based approaches, seem to be more adapted for ICS. In behavioral approach, specifications of the system are considered to define rules to insure safety, reliability and security of the process. Even if a lot of publications are based on network analysis, ICS are used to control process with physical sense and some works focus on this aspect. Interesting use of process knowledge can be found in (Carcano et al., 2011; Fovino et al., 2012). After modeling the process by setting ranges of running for each variable, the authors use the concept of distance from critical state to detect attacks. Evaluating the distance defines the increasing closeness between safe state and critical area. 2 types of distance are used. d_1 computes the gap between 2 states component by component. Let d_1 be any notion of distance between two states as d_1: $\mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^+$, s and t are two vectors with n components as $s \in \mathbb{R}^n$ and $t \in \mathbb{R}^n, d_1(s,t) = \sum_{i=1}^{n} |s_i - t_i|$. d_v counts difference between 2 vectors based on the number of different components. Let d_v be any notion of distance between two states as d_v: $\mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^+$, s and t are two vectors with n components as $s \in \mathbb{R}^n$ and $t \in \mathbb{R}^n, d_v(s,t) = \#\{i,\ s_i \neq t_i\}$.

The study focuses on approaches with strong behavioral approach, or model based, by introducing the concept of distance. Finally, IDS probe structure which intercepts and analyzes data, is similar with filter structure in filter approach.

3. FILTER APPROACH

First, the filter approach was enunciated by (Cruette et al., 1991), originally intended for applications in risk assignment. This approach presents interesting points for cybersecurity of ICS. The basis of the method is perpetual evaluation of all data between *Command Part* or CP (control layer, level 1) and *Operative Part* or OP (low level layer, level 0). If work order transgresses constraints in the filter, then an alert is sent to operator and the information may be stopped. A command filter can stop an incorrect order to be sent to actuators whereas report filter alerts on data between CP and OP. (Marangé, 2009) continued this work with a formalization of the constraints inside the filter. This approach only deals with a command filter because it is considered that operative part cannot have a hardware failure. Data from PLC to actuators are then filtered based on use constraints not to reach forbidden states. These constraints are enunciated by an expert and are static (physical impossibility) or dynamical (prohibition of an event based on a set of other events). Fig. 2 illustrates the running of the filter approach.

The approach is an interesting solution with several advantages as evaluating the accuracy of the information exchanged before it is executed by actuator for command filter or taken into account by CP for report filter. The assessment is performed by rules based on the knowledge of the process and control law. Finally, the approach is rather non-intrusive in the control command architecture. The study will improve rules establishment, especially with the concept of state to ensure that experts do not forget rules. Moreover, filter structure allows implementation of detection algorithm as well as failure origin discrimination. The notion of distance tested in IDS field is used similarly in filter approach. During the study, filters are considered as not vulnerable to an attack.

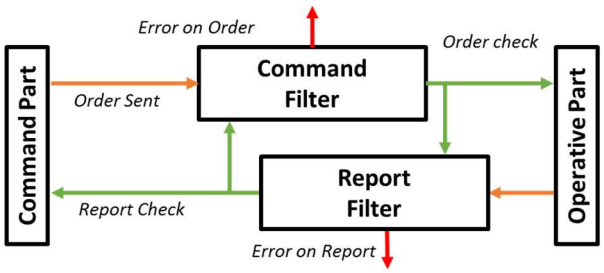

Fig. 2. Illustration of the filter approach

4. METHOD OF FILTER CONCEPTION

Intrusion detection with filter approach is efficient only if rules implemented into the filter are characteristics and

representatives of the monitored system. Three steps are required to build the filter. First, risk assignment step allows to identify unsafe states, which damage the system, from others. Then state exploration step, which is based on the description of actions, finds the relation between each state and the combinations that lead to unsafe state. Finally, synthesis filter step implements detection rules and distance algorithm. These steps are further elaborated in following subsections.

4.1 Step 1: Risk assignment

The first step is based on the work done in risk assignment that is to verify the ability of an entity to satisfy one or more required function under given conditions (Fourastier and Pietre-Cambacedes, 2015). Information can be extracted to identify failure modes and determine causes leading to these states. Indeed, when an ICS is under attack, hackers will always try to degrade the system by bringing it to an unsafe state with actions that should not have been made at this time. Such system can only be in a unique type of state as presented in Fig. 3 and described below.

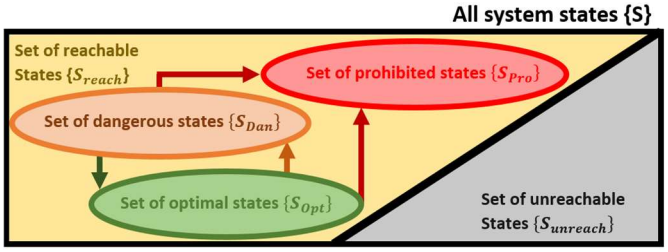

Fig. 3. Illustration of possible states in an ICS

A state $S_i \in \{S\}$ of a system can be reachable or not. For an ICS, state vector $S_i \in \{S_{reach}\}$ is the representation of the process and data sent to the PLC at particular moment, such that S_i is a vector of n components where n is the number of inputs. In the same way, an order O_j, in the set of orders $\{O\}$ and $O_j \in \{O\}$ is an action sent to the OP. It is a vector of m components where m is the number of outputs. The notion of state in the process was introduced by (Mitchell and Chen, 2014). A reachable state can be in one the following subsets:

- Optimal state S_{Opt}. $\{S_{opt}\} \subset \{S_{reach}\}$ respects the proper running of the process and the constraints imposed by PLC. PLC programming is oriented to always keep the system in this subset of safe states,
- Dangerous state S_{Dan}. $\{S_{Dan}\} \subset \{S_{rea}\}$ violates the constraints imposed by the control law without causing critical degradations on the process,
- Prohibit state S_{pro}. $\{S_{pro}\} \subset \{S_{reach}\}$ does not respect constraints of the control law and results in a significant degradation of the process and its environment. The subset has to be avoided and the system has to be stopped before being in S_{pro}.

Finally, O_{Opt} is the order respecting the control law, S_{pre} and O_{pre} are respectively the state and the order predicted by filters.

4.2 Step 2: System state exploration

The objective of this step 2 is to explore the different evolutions of the system, to categorize them in the three possible states and to list the prohibit states of the system. Control law model and process model are used. The first one organizes all the actions to be done and the schedule in order to reach the final state of the process in the fastest and safest way. Process model details all reachable states based on the description of all actions and their effects on the system.

Control model is obtained directly from the control law inside of the PLC. Modeling, used later to find the optimal states, is based on Petri-net model which allows expressing constraints of a control law easily, especially sequence properties (task scheduling), parallelism (execution of several tasks at the same time), mutual exclusion (execution of an activity prevents execution of others) and synchronization (waiting the end of one, or several, specific activity before executing others). Regarding process model, a finite number of actions in ICS can be executed by the control part on the operative part so that the number of states is also limited. The effect on OP of each action of the system can be described. Modeling with a deterministic finite automaton is adapted to describe this procedure. An automaton M, representative of process capacities contained in variable δ and obtained by using (Henry et al., 2012), is defined by the following quintuplet:

$$M = \{\{\sum_{i=1}^{n} S_i\}, \{\sum_{j=1}^{m} O_j\}, \delta, S_0, S_{final}\}$$

- $\{\sum_{i=1}^{n} S_i\}$ is the finite set of possible states where n denotes numbers of reachable states. S_i is a vector representing the state of the process which can be $\{S_{Opt}\}, \{S_{Dan}\}$ or $\{S_{pro}\}$,
- $\{\sum_{j=1}^{m} O_j\}$ is the finite set of action that can be performed by the system,
- δ is the relation between states and an order. These relationships will make change the automaton by simulating the behavior of the system. Each action has to be modeled by its effect on the process,
- $S_0 \in \{\sum_{i=1}^{n} S_i\}$ is the initial state of the system,
- $S_{final} = \{S_0, \{S_{pro}\}\} \subset \{\sum_{i=1}^{n} S_i\}$ is the set of final states. This may be the initial state of the system S_0 or a state in which no further evolution of the process is possible (forbidden state $\{S_{pro}\}$).

Algorithm, presented in Fig 4. and based on this automaton, applies for one given state S_i one action $O_j \in \{O\}$ to discover all the reachable state for the system $\{S_{reach}\}$. If the projection leads to a forbidden state $S_{i+1} \in \{S_{pro}\}$ then the context, composed of state S_i and action O_j is saved. Moreover, if the new state S_{i+1} has already met earlier in the algorithm then the state is removed from the list of states that will be evaluated by the algorithm during the next iteration. The same thing happens if the projected state is the initial state or a prohibited state. The next iteration will only explore states not found previously. As a result, processing speed is increased,

combinatory explosion is prevented and a unique branch is explored every time.

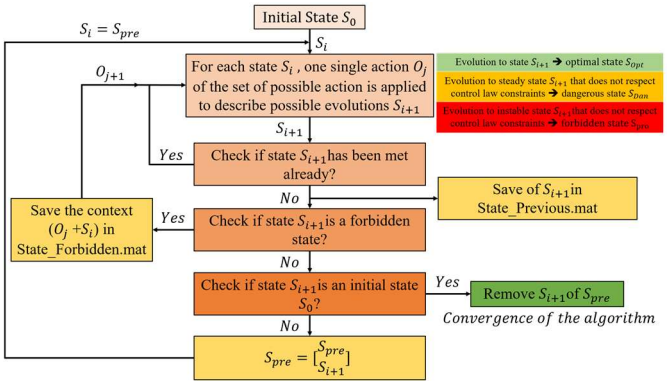

Fig. 4. Illustration of the exploration states algorithm

4.3 Step 3: Synthesis filter implementation

The rules describing constraints of the process are implemented into command filter in this third step. Process modeling and the list of context leading to prohibited states are used to write a rule $R = \{\sum_{k=1}^{l} R_k\}$. When:

$$S_i \in \{S|S_{pro}\}, O_j \in \{O\}\ s.t.\ \delta(S_i, O_j) \in S_{pro}$$

Rules are described as a matrix where:

$$\begin{cases} \forall (a,b) \in \mathbb{N} \\ 1 \le a \le n \\ 1 \le b \le m \end{cases}, R = (r_{a,b}) = true \ \leftrightarrow\ \delta(S_a, O_b) \notin \{S_{pro}\}$$

Rules are a static security measure. The command filter in which they will be implanted, evaluates the common context of the system and compares it to these rules. If the context respect these rules, then order O_j is sent to OP. However, the convergence of ICS to unsafe state could be anticipated and quantified by computing distance between common state and prohibited states or optimal states. Indeed, in case of attack, a hacker will always try to bring the system in a prohibited state S_{pro} to make it inoperative. In such a case, distance between common state and forbidden state will reduced to 0. In contrary to a material failure, the distance is more stable. By applying distance concept of (Carcano et al., 2011) with the model of the process, distance between common and forbidden states are dynamically computed. However, more deviation can be detected considering control model. An attack, changing orders sent to OP but having the same effect on the process, is anticipated. A distance vector d taking into account the two models (control and process) is proposed:

$$d \in \mathbb{R}^+, d_1 \in \mathbb{R}^+, d_v \in \mathbb{R}^+, (i,j) \in \mathbb{N}\ and\ (s,t,o,p) \in \mathbb{R}^n$$

$$d = \begin{cases} d_1 = \sum_{i=1}^{n} |s_i - t_i| + \sum_{j=1}^{m} |o_j - p_j| \\ d_v = \#\{i,\ s_i \neq t_i\} + \#\{j,\ o_j \neq p_j\} \end{cases}$$

The vector allows quantifying distance between common state and optimal or prohibited states of the system at t time. By monitoring the evolution of the distance, trajectory concept can be established. The notion not only allows detection but also discrimination of an attack from a failure. In order to guarantee duplication with contexts found in step 2, functions were implemented into the command filter allowing the prediction of future system state based on common state and sent order. Then, distance between common state and expected optimal state and the nearest forbidden state is computed. Finally, context is tested to analyze if it respects rules of the system. For report filter, only distance computation algorithm is implanted in the filter. Distance assessment (between common and expected states) is sent back to operators.

5. SIMULATION EXAMPLE

To illustrate the study, a well-known example on the literature as provided in (Li et al., 2016) is used. The system is composed of three tanks. Two tanks T_1 and T_2 of infinite capacity, contain respectively product A and product B. Each tank discharges their product into a melting tank T_3 in order to produce a product C. The filling stage is done by opening valves V_1 and V_2. Level sensors show the effect of actuators on the process. When sensor H_2 is activated, valve V_3 opens to drain T_3. 3 level sensors $H_0(LL), H_1(H)\ and\ H_2(HH)$ are used to adjust the opening of the valves. There is only 1 forbidden state for the system which is reached when level in tank T_3 exceeds H_2.

State vector $[Sensor\ H_0, Sensor\ H_1, Sensor\ H_2]$with all or none sensors, order vector $[Valve\ V_1, Valve\ V_2, Valve\ V_3]$, step state vector $[S_{filing}, S_{draining}]$ are defined. During state exploration step, there are 8 different states:

- Unattainable states set $\{S_{unreach}\}$: $\{[0\ 1\ 0], [0\ 0\ 1], [1\ 0\ 1], [0\ 1\ 1]\}$,
- Reachable states set $\{S_{reach}\}$: which contains $\{S_{Opt}\}$: $\{[0\ 0\ 0], [1\ 0\ 0], [1\ 1\ 0], [1\ 1\ 1]\}$, $\{S_{Dan}\}$: $\{\emptyset\}$, $\{S_{pro}\}$: $\{S_{overflow}\}$.

Contexts leading to forbidden states $\{S_{pro}\}$ are found by the algorithm and are regrouped in Table 1. The automaton M representing the running of the system is:

$$M = \{\{\sum_{i=1}^{13} S_i\}, \{\sum_{j=1}^{8} O_j\}\ , \delta, S_0, S_{final}\}$$

With $\{\sum_{i=1}^{13} S_i\}$ the set of reachable states composed of $\{S_{Opt}\}$ coupling with step state vector and $\{S_{overflow}\}$, $\{\sum_{j=1}^{8} O_j\}$ the set of orders made up of every possible combinations of order vector, initial state S_0 : [0 0 0], final state S_{final}: $\{S_0, S_{pro}\}$ and δ the relationship between states and orders. For example:

$$\forall (S_a, S_b) \in \{\sum_{i=1}^{13} S_i\} \setminus \{S_{pro}\},\ O_3 = [0\ 0\ 1],\ \delta(S_a, O_3) = S_0$$

$$\forall k \in [1,2]\ st \begin{cases} S_a(1,k) = 1 \\ S_a(1,k+1) = 0 \end{cases}, S_a = [1\ 0\ 0], O_1 = [1\ 0\ 0],$$

$$\delta(S_a, O_1) = S_b with\ S_b(1,k+1) = S_a(1,k+1) + 1$$

Algorithms presented in previous sections are implemented depending on the location of the filter in the control-command architecture. An attack "Man in the Middle" is simulated. When the attack is launched, all the orders are intercepted and replaced by predefine orders. In our example, orders $O_j \in O_{Opt}$ are replaced by order $O_{attack} \in \{O\}$ and replaying the first order of the sequence $O_1 = [1\ 0\ 0]$ that opens valve V_1. During the first cycle, order O_{attack} is sent and opens valve V_1 and fill

in tank T_3. Command filter detects no deviations on the command model and on the process model because attack order corresponds to optimal order O_1. The process goes from optimal state (initial state) to another one, distance to optimal states is equal to 0. When, sensor level H_1 is activated, the order to open V_2 is sent and is replaced by O_{attack}. Command filter detects a deviation from control law model $O_{attack} \neq O_{Opt}$ but the predicted state match with optimal state $S_{Pre} = S_{Opt} = [1\ 1\ 0]$. Distance from optimal is 1 and the one from forbidden state is also 1. The order is sent because context does not match with ones that lead to prohibited states. In consequence, level sensor H_2 is activated and an alert is communicated to the supervision because optimality of the process is not respected (V_1 was opened instead of V_2). Finally, when the draining has been done, O_{attack} is sent to the ICS. Command filter detects another deviation. Then, context leads to a forbidden state (tank overflow). As one rule is broken, the distance from prohibited states is equal to 0 and order O_{attack} is stopped by the filter. Physical damaged on the system and its environment are avoided.

Table 1. Risk assignment analysis

Forbidden States $\{S_{pro}\}$	Order O_j	State of the process before the action S_i
Case 1: only one order		
Tank Overflow	Open V_1 or V_2 [1 0 0] or [0 1 0]	Sensor H_2 activated: [1 1 1]
Case 2: multiple orders		
Tank Overflow	Open V_1 or V_2: [1 0 0] or [0 1 0]	Sensor H_2 activated: [1 1 1]
	Open V_1 and V_2: [1 1 0]	Sensor H_2 activated: [1 1 1]
	Open V_1 and V_2: [1 1 0]	Sensor H_1 activated: [1 1 0]

6. CONCLUSIONS

In the paper, a new approach, based on level 1 and 0, for detecting cyberattacks against industrial installations is presented. The keystone of the approach is the notion of distance between different states. The distance is a new mean to detect an attack based on the assumption that the purpose of an attacker purpose is to lead a system into prohibited state. The notion of distance introduces the concept of trajectory which is based on the evolution of distance over time, discriminates whether ICS are faced to a complex and intentional cyber-attack or not.

The approach detects attacks based on the knowledge of physical process with models. These models lead to rules that prevent dangerous orders execution. The concepts of distance and trajectory will be improved to discriminate detections. Finally, the results conducted on simulation demonstrate the feasibility of the approach and validate the proposed approach. More tests will be conducted on several prototypes representing real industrial systems.

REFERENCES

Carcano, A., Coletta, A., Guglielmi, M., Masera, M., Nai Fovino, I., Trombetta, A., 2011. A Multidimensional Critical State Analysis for Detecting Intrusions in SCADA Systems. IEEE Trans. Ind. Inform. 7, 179–186. doi:10.1109/TII.2010.2099234

Caselli, M., Zambon, E., Kargl, F., 2015. Sequence-aware Intrusion Detection in Industrial Control Systems, in: Proceedings of the 1st ACM Workshop on Cyber-Physical System Security. ACM, New York, NY, USA, pp. 13–24. doi:10.1145/2732198.2732200

Cruette, D., Bourey, J.P., Gentina, J.C., 1991. Hierarchical specification and validation of operating sequences in the context of FMSs. Comput. Integr. Manuf. Syst. 4, 140–156.

Denning, D.E., 1987. An Intrusion-Detection Model. IEEE Trans. Softw. Eng. SE-13, 222–232. doi:10.1109/TSE.1987.232894

Falliere, N., Murchu, L.O., Chien, E., 2011. W32. stuxnet dossier. White Pap. Symantec Corp Secur. Response 5, 6.

Fourastier, Y., Pietre-Cambacedes, L., 2015. Cybersécurité des installations industrielles : défendre ses systèmes numériques. Cépaduès Editions.

Fovino, I.N., Coletta, A., Carcano, A., Masera, M., 2012. Critical State-Based Filtering System for Securing SCADA Network Protocols. IEEE Trans. Ind. Electron. 59, 3943–3950. doi:10.1109/TIE.2011.2181132

Henry, S., Zamaï, E., Jacomino, M., 2012. Logic control law design for automated manufacturing systems. Eng. Appl. Artif. Intell. 25, 824–836. doi:10.1016/j.engappai.2012.01.005

ICS-CERT, 2016. ICS-CERT / The Industrial Control Systems Cyber Emergency Response Team [WWW Document]. URL https://ics-cert.us-cert.gov/

Li, W., Xie, L., Deng, Z., Wang, Z., 2016. False sequential logic attack on SCADA system and its physical impact analysis. Comput. Secur. 58, 149–159. doi:10.1016/j.cose.2016.01.001

Marangé, P., 2009. Synthèse et filtrage robuste de la commande pour des systèmes manufacturiers sûrs de fonctionnement. Université de Reims.

McLaughlin, S., Konstantinou, C., Wang, X., Davi, L., Sadeghi, A.-R., Maniatakos, M., Karri, R., 2016. The Cybersecurity Landscape in Industrial Control Systems. Proc. IEEE 104, 1039–1057. doi:10.1109/JPROC.2015.2512235

Mitchell, R., Chen, I.-R., 2014. A survey of intrusion detection techniques for cyber-physical systems. ACM Comput. Surv. 46, 1–29. doi:10.1145/2542049

Pan, S., Morris, T.H., Adhikari, U., Madani, V., 2013. Causal Event Graphs Cyber-physical System Intrusion Detection System, in: Proceedings of the Eighth Annual Cyber Security and Information Intelligence Research Workshop. ACM, New York, NY, USA, p. 40:1–40:4. doi:10.1145/2459976.2460022

Wang, Y., Xu, Z., Zhang, J., Xu, L., Wang, H., Gu, G., 2014. SRID: State Relation Based Intrusion Detection for False Data Injection Attacks in SCADA, in: Kutyłowski, M., Vaidya, J. (Eds.), Computer Security - ESORICS 2014. Springer International Publishing, pp. 401–418.