Computers & Security 125 (2023) 103010

Contents lists available at ScienceDirect

Computers & Security

journal homepage: www.elsevier.com/locate/cose

Forensic readiness of industrial control systems under stealthy attacks

Mazen Azzam[a,*], Liliana Pasquale[b], Gregory Provan[c], Bashar Nuseibeh[a,d]

[a] Lero @ the University of Limerick, Limerick V94 T9PX, Irelandk
[b] Lero @ the University College Dublin, Belfield, Dublin 4, Ireland
[c] Lero @ the University College Cork, College Road, Cork T12 K8AF, Ireland
[d] The Open University, Milton Keynes MK7 6AA, UK

ARTICLE INFO

Article history:
Received 11 July 2022
Revised 1 November 2022
Accepted 7 November 2022
Available online 17 November 2022

Keywords:
Industrial control systems
Forensic readiness
Digital forensics
Safety checking
Stealthy attacks
Value of information

ABSTRACT

Cyberattacks against Industrial Control Systems (ICS) can have harmful physical impacts. Investigating such attacks can be difficult, as evidence could be lost to physical damage. This is especially true with *stealthy attacks*; i.e., attacks that can evade detection. In this paper, we aim to engineer *Forensic Readiness* (FR) in safety-critical, geographically distributed ICS, by proactively collecting potential evidence of stealthy attacks. The collection of all data generated by an ICS at all times is infeasible due to the large volume of such data. Hence, our approach only triggers data collection when there is the possibility for a potential stealthy attack to cause damage. We determine the conditions for such an event by performing predictive, model-based, safety checks. Furthermore, we use the geographical layout of the ICS and the safety predictions to identify data that is at risk of being lost due to damage, i.e., relevant data. Finally, to reduce the control performance overhead resulting from real-time data collection, we select a subset of relevant data to collect by performing a trade-off between expected impact of the attack and the estimated cost of collection. We demonstrate these ideas using simulations of the widely-used Tennessee–Eastman Process (TEP) benchmark. We show that the proposed approach does not miss relevant data and results in a reduced control performance overhead compared to the case when all data generated by the ICS is collected. We also showcase the applicability of our approach in improving the efficiency of existing ICS forensic log analysis tools.

1. Introduction

In Industrial Control Systems (ICS), software handles safety-critical processes that often form the core of a nation's critical infrastructure; e.g., power generation and chemical processes. Differently from traditional IT systems, cyberattacks against ICS can cause physical damage (Lee et al., 2014). Following an attack, a digital forensics investigation is usually performed to identify how an incident occurred using digital evidence (Carrier and Spafford, 2004). However, digital evidence in ICS can be volatile, as control devices typically feature low storage resources and can be damaged by attacks (Van Vliet et al., 2015). ICS therefore need to be *Forensic-Ready* (Kebande and Choo, 2022; Rowlingson, 2004; Zainudin et al., 2022), i.e., capable of identifying, collecting, and preserving, *in advance*, data that may be used as evidence to investigate potential, known incidents (Alrajeh et al., 2017).

The collection of all data at all times may not be feasible due to the large amount of data typically generated by an ICS. This is particularly true for large-scale, geographically-distributed ICS; i.e. those that consist of a large number of process equipment and control devices whose layout occupies a large area — e.g. chemical processes and power plants. Data stored in an ICS' process historian, while potentially useful in diagnosing anomalies, may not explain *how* an attack occurred (Ahmed et al., 2017; Janicke et al., 2015). Most of the work in ICS forensics is geared towards post-incident investigations (Mohamed et al., 2020) and a few approaches attempt to investigate potential attacks while the system is running, performing *live forensics* (Taveras, 2013).

Previous work relies on the detection of attacks to trigger data collection. Thus, it may not be effective when an ICS is targeted by a *stealthy attack*, which can take advantage of sensor noise or other physical properties of the system to evade anomaly detectors (Giraldo et al., 2018). Also, previous work does not focus on the collection of *relevant* data (Pasquale et al., 2018) that can explain how an attack occurred and can also be lost to damage caused by an attack. The real-time collection of such data from

* Corresponding author.
E-mail addresses: mazen.azzam@lero.ie (M. Azzam), liliana.pasquale@ucd.ie (L. Pasquale), g.provan@cs.ucc.ie (G. Provan), bashar.nuseibeh@lero.ie (B. Nuseibeh).

https://doi.org/10.1016/j.cose.2022.103010

control devices can incur a performance overhead, which can negatively affect the performance requirements for safe control operations. In this paper, we aim to engineer Forensic-Ready, safety-critical, geographically distributed ICS, which can proactively collect relevant data to a stealthy attack. To achieve this aim, we need to identify a *trigger* to data collection for stealthy attack and a technique to identify *relevant* data, while reducing any performance overhead.

To trigger data collection, we rely on previous work on online safety monitoring under stealthy attacks (Azzam et al., 2021; 2022; Kwon and Hwang, 2018). Instead of forcing the detection of such attacks, online safety monitoring asks whether a *potential* stealthy attack can cause damage to the system given an initial physical state. While this may not reveal such attacks, it can identify conditions under which a system can be damaged and, thus, relevant data should be collected. To identify *relevant* data which is also at most risk of being damaged, we propose a technique that relies on the geographical layout of the system, safe Process Plant Layout (PPL) (Quiroz-Pérez et al., 2021). To reduce the real-time data collection overhead, we propose a decision-theoretic framework to decide whether the identified data is "worth" collecting based on a trade-off between collection cost and the expected damage that can be caused by the attack.

We show through extensive simulations on the benchmark Tennessee–Eastman Process (TEP) that our approach does not miss any relevant data; and the collection of data enabled by our decision-theoretic framework has a limited impact on the controllers' performance (execution time). Additionally, we demonstrate a use case of our approach where the performance of Programmable Logic Controller (PLC) log analysis tools (Yau and Chow, 2015) can be significantly improved. The rest of the paper is organised as follows: Section 2 overviews related work; Section 3 clarifies the problem statement and summarizes the overall approach; and Section 4 describes a running example. We detail our approach in Section 5 and evaluate it in Section 6, before concluding the paper in Section 7.

2. Related work

Differently from traditional IT systems, attacks on ICS may seek to cause physical damage rather than stealing or tampering with confidential data. Due to their effect on the behaviour of physical processes, a large body of work has considered techniques from control engineering to detect attacks on ICS (Giraldo et al., 2018). It has been shown that resourceful attackers can exploit noise (Griffioen et al., 2019) and other control theoretical properties (Pasqualetti et al., 2013) to evade anomaly detectors. The detection of these so-called stealthy attacks still faces theoretical and practical limitations, such as interference with the safety-critical operation of ICS devices (Griffioen et al., 2019).

Forensics in ICS faces the following challenges. The safety criticality of industrial processes and the difficulty of shutting them down for investigations lead to strict constraints on forensic tools to limit interference with their operation. The lack of documentation in most ICS devices can complicate data acquisition tasks and may lead to a loss of data, as described by Van Vliet et al. (2015). In addition, the limited resources available in low-level devices (e.g., PLC's, sensors, actuators) Awad et al., 2018; Eden et al., 2016; van der Knijff, 2014) renders potential evidence volatile (Eden et al., 2016; Mohamed et al., 2020). Therefore, ICS forensics requires different approaches than traditional IT systems. Most of the existing work in ICS forensics attempts to adapt traditional IT forensic investigation frameworks to apply them to ICS (Altschaffel et al., 2019; Elhoseny et al., 2017; Mishra, 2019). Other work has proposed techniques to perform live forensics, i.e., investigate potential attacks while the system is running (Al-Sharif et al., 2018; Taveras, 2013). In addition, several approaches have focused on analysing and acquiring evidence from specific ICS devices, namely PLC's (Chan et al., 2018; Qasim et al., 2020; Yau et al., 2018) and engineering workstations (Hadžiosmanović et al., 2012; Myers et al., 2017).

Most of existing work in ICS forensics has focused on forensic investigations undertaken after an incident or an anomaly has occurred, as in the work of Taveras (2013). A limited body of work has considered proactive approaches to ICS forensics. Examples include the work by Grispos et al. (2017) and Ab Rahman et al. (2016), which consider Forensic Readiness (FR) requirements in the design of medical and cloud-based CPS, respectively. However, these approaches have focused on generic organisation-level guidelines for pre-planning incident response measures and cannot be applied to safety-critical ICS.

Our work considers engineering FR in safety-critical ICS faced with stealthy attacks, which can cause physical damage to an ICS and lead to the loss of potential evidence. Although live forensics techniques (Taveras, 2013) aim to collect data proactively, the collection of such data is only triggered after an attack is detected. Therefore, these techniques may not be effective when an ICS is targeted by a stealthy attack. Our work has a similar aim as the approach proposed by Alrajeh et al. (2017), which generates data collection specifications to support forensic readiness in a traditional IT system. However, to the best of our knowledge, our work is the first to address the challenges that stealthy attacks pose to ICS forensics.

Our approach is based on a physics-based Early Warning System (EWS) proposed by Azzam et al. (2021, 2022) to trigger potential evidence preservation in ICS. The approach proposed by Azzam et al. (2021, 2022) allows the real-time identification of conditions under which there is a possibility for a stealthy attack to cause damage. While the proactive collection of data that may represent potential evidence is suggested as possible post-warning measure, the authors do not suggest which data should be collected. Our approach, instead, proposes a technique to identify relevant data that should be collected based on the PPL and a formal decision-theoretic framework.

3. Problem statement and overview

In this section, we clarify the problem tackled in this paper, and provide an overview of our approach.

3.1. Problem statement

In this paper, we seek to collect data that could explain whether and how a stealthy attack occurred. We also aim to collect this data before it may be lost due to the damage caused by the attack. Our problem has three main parts (P1–P3). *(P1)*: Since stealthy attacks can avoid detection, it is not possible to use the alarms generated by anomaly detectors to trigger proactive data collection or live forensics (Taveras, 2013). Thus, it is necessary to identify an alternative trigger for proactive data collection. *(P2)*: not all data generated by an ICS may require to be collected proactively, since only a subset of such data may be at risk of being lost due to damage. As such, there is a need for a technique to identify what data is faced with such a risk. *(P3)*: ICS generally operate under strict real-time performance requirements and are expected to generate a profit; thus, data collection activities can negatively affect the performance of the plant. To reduce the overhead introduced by data collection, this activity should only be performed when the cost of collection does not exceed the potential damage that can be caused by a stealthy attack.

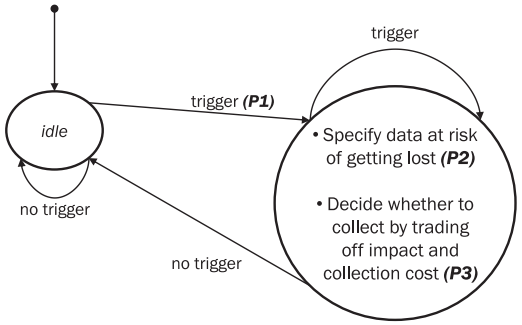

Fig. 1. An automaton model of the proposed approach to support FR in ICS.

3.2. Overview of the approach

Our approach to support FR in ICS presented with stealthy attacks consists of three main steps, which address the problems (P1-P3).

1. To trigger data collection (P1), we rely on predictive safety checks of whether a potential stealthy attack can indeed take the system into an unsafe state and potentially cause the loss of data.
2. To determine what data is at the risk of being lost due to the potentially damaging stealthy attack (P2), i.e., relevant data, we employ the safe PPL (Quiroz-Pérez et al., 2021) to identify areas of the plant that may be affected by the suspected breach. Within these areas, we identify devices and data originating from these devices that can represent potential evidence and that are at risk of being damaged.
3. Among the set of data that may potentially be lost, we identify the ones should be collected (P3). To achieve this aim, we propose a decision-theoretic framework which decides to collect data only if the expected impact of the attack, in terms of its potential effect on the system's performance, does not exceed the cost of collection. Accordingly, we collect a subset of relevant data which results in a limited overhead on the system's real-time performance.

Fig. 1 shows the three steps of our approach in the form of an automaton reflecting the real-time deployment of the proposed approach. Our main assumption in this work is that data acquisition tools, such as the one suggested by Yau and Chow (2015), are available to be activated whenever data should be collected from a device of an ICS. In addition, we assume that a secure server is available to store the collected data.

4. Running example

We use a chemical plant setup based on the benchmark Tennessee–Eastman Process (TEP) as a running example to describe the proposed approach. We begin this section with a brief description of the TEP setup, and then describe our attack scenario example.

4.1. Tennessee–Eastman setup

The TEP is a benchmark chemical process suggested by Downs and Vogel (1993) and based on a real industrial plant that produces two liquid products from four gaseous reactants. The TEP has been recently used as a virtual testbed for various works in ICS security as it represents a realistic chemical processing environment. The process consists mainly of five major operating units: a reactor vessel, a product condenser, a vapour-liquid separator, a recycle compressor, and a product stripper. For the purposes of our work, we assume that the system layout is as shown in Fig. 2. Each main operating unit in the TEP is housed in its own control room, which contains - aside from the process equipment - low-level control devices such as sensors/actuators, PLCs, Remote Terminal Units (RTUs) and engineering workstations. Each control room is connected over a network to a central room that houses supervisory control servers and process historians. In each control room, the relevant controllers' logic is installed on the PLCs which are in turn connected to the engineering workstations via an RTU. The RTUs relay control data to the central control room, as in the case with Supervisory Control and Data Acquisition (SCADA) architectures. Furthermore, an anomaly detector located in the central control room monitors control data, namely sensor measurements and actuator inputs, to detect anomalies due to faults or malicious tampering.

4.2. Attack scenario

Our example is inspired by the Stuxnet (Albright et al., 2011) and the German steel mill (Lee et al., 2014) attacks. Namely, we consider the reactor stage of the TEP (area A_1) which features temperature and pressure controllers that keep these operating variables at desired levels. We assume that each of these controllers is installed on a Siemens S7 PLC, which is connected to sensors and actuators using a network that employs a standard industrial *Profinet* ethernet. Due to control safety constraints, communications happening using the TCP/IP protocol are performed in plain text without any encryption (Ahmed et al., 2017).

In this scenario, the attacker gains access to a network via a social engineering attack involving a fraudulent email that includes a malicious attachment. The attack then proceeds as following: in Phase (1), the reconnaissance phase, the attacker identifies some properties of the PLC, where temperature and pressure controllers are implemented. These properties can include the make, model, firmware, function codes, and addresses of the PLC (Ahmed et al., 2017). Due to the lack of encryption, the attacker sniffs the data exchanged between the PLC and the physical system and uses it to extract more knowledge about the system, such as its noise properties, and details about the physics-based anomaly detector in use.[1]

In phase (2), and to avoid detection, the attacker chooses to slowly drive the reactor to an unsafe state before any anomaly can be detected in the central station. To this end, they exploit the PLC function codes identified previously to modify the pressure control program installed on the PLC. This modification involves applying a slowly growing bias to pressure measurements received by the PLC, such that the controller is tricked into increasing the pressure in the reactor to unsafe levels. The attacker can hide these deviations from the anomaly detector using their knowledge of the system dynamics and measurement noise properties (Murguia and Ruths, 2018) that they established during the reconnaissance phase.

A forensic investigation into the incident will likely need to recover potential evidence from the equipment located near the reactor, namely the engineering workstation, PLC, and RTU. This however may not be possible, as a pressure buildup in the reactor may lead to an explosion, thus, severely damaging these devices. For example, Van Vliet et al. (2015) describe the difficulties of recovering data from a PLC that was damaged in a wind turbine fire. Thus, this data need to be collected proactively.

ICS are typically equipped with a process historian — a proprietary server that records process data over long periods of

[1] We assume that the attacker has a vast amount of resources at their disposal to perform such activities. In the case of Stuxnet, it is generally agreed that the attackers had access to such resources (Lee et al., 2014).

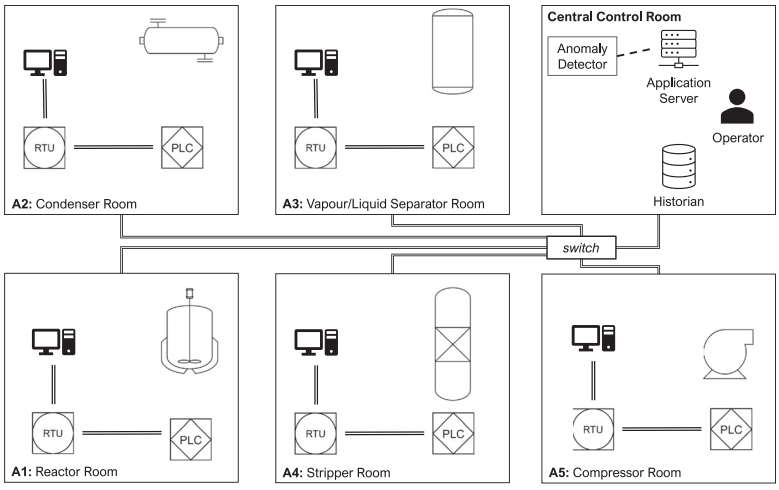

Fig. 2. Layout of the TEP plant used in the running example.

time. Although data recorded in process historians are protected from damage, it is geared towards supervising the physical process rather than detecting attacks (Ahmed et al., 2017; Altschaffel et al., 2019; Janicke et al., 2015). In the TEP scenario for example, a process historian may contain pressure measurements received from the reactor over a certain period of time. These measurements may reveal potential anomalies caused by an attacker. However, a forensic investigator may not be able to infer that an attacker performed code modifications to the PLC without having access to the PLC configuration at the time. Such data is not typically stored in process historians but it can be collected using our approach.

5. Approach to proactive data collection

We begin this section with a description of the kind of stealthy attacks that we focus on, before detailing our approach (Fig. 1) to engineering FR in ICS. We then compile the steps of our approach in a real-time data collection algorithm.

5.1. Stealthy sensor attacks

We focus in this work on attacks that consist of modifying the behaviour of the physical process in ICS. In particular, we consider stealthy attacks that consist of slowly modifying sensor measurements while exploiting sensor noise to evade detection. We assume that the attacker, as illustrated in Section 4, has knowledge of the system's model and anomaly detection procedure. Additionally, the attacker is able to alter sensor measurements either by modifying controller code or by accessing relevant nodes in the network. Regardless of the manner by which the attacker performs their actions, our approach only considers the effect of such actions on the physical process.

Stealthy sensor attacks have been widely considered in previous work (Giraldo et al., 2018). We choose to focus on them due to their practicality from the point of view of an attacker and the ease of maintaining their stealthiness as compared with attacks on actuators (Urbina et al., 2016). Furthermore, the ICS that we consider may experience long operational transients, which renders replay attacks more difficult to keep stealthy (Azzam et al., 2022). A model for stealthy sensor attacks can be found in the work of Murguia and Ruths (2018).

5.2. Trigger for real-time data collection (P1)

Real-time data collection activities can be triggered following an alarm from an existing Anomaly Detection System (ADS) (Taveras, 2013). However, in the case of the stealthy attacks we are concerned with, such a trigger may not be available as the resourceful attacker may evade the existing ADS and may be able to cause damage before any such alarm is raised.

To mitigate this problem, we propose instead a trigger that relies on online safety monitoring under attacks. Instead of attempting to detect stealthy attacks, this line work seeks to proactively check in real-time for conditions under which a *potential* stealthy attack can cause damage to the system (Azzam et al., 2021; 2022; Kwon and Hwang, 2018). This can be useful to guide certain proactive measures, including the collection of potential evidence. Of the approaches proposed in the literature, we adopt the one proposed by Azzam et al. (2021, 2022) where safety checks are used to compute a so-called *suspicion metric* reflecting the probability of damage taking place due to a stealthy attack.

The *suspicion metric* combines two physics-based preliminary indicators to warn in advance of potential damage: (i) *feasibility* of a stealthy attack, defined as its ability to drive the system into the unsafe operating region without getting detected; and (ii) *proximity* of the system to unsafe operating region. These two indicators are then expressed in terms of a reachability problem, where reachable sets are approximated using ellipsoids computed through a linear matrix inequalities setup. The size of the reachable set and of its intersection with a predefined set of unsafe states, as well as the approximate time to reach the unsafe set are then combined to compute the suspicion metric. Due to space limitations, we refer the reader to the authors' work (Azzam et al., 2021; 2022) for more details on their approach.

The reasons for adopting the approach by Azzam et al. (2021, 2022) are as follows: (1) The online safety monitoring algorithm is efficient and can scale well with complex systems. (2) The algorithm is tailored to systems that can be modelled as Linear Time-Invariant (LTI) which is a widely used modelling paradigm in practice, and can be used to model the TEP in our running example. (3) The probability of damage, given by the suspicion metric, can be used to estimate the impact of the suspected attack.

The approach by Azzam et al. (2021, 2022), which is already instantiated for the TEP, can raise warnings having two different levels of criticality, based on the likelihood of damage taking place given by the suspicion metric. The first is a low-criticality warning where the system may reach an unsafe state if it is under a stealthy sensor attack, but is far from these states. The second is a high-criticality warning where the system is dangerously close to the unsafe operating region. The first level of warning suggests that there is sufficient time to collect potential evidence before a

Table 1
Potentially relevant data (for the stealthy attack that we consider) at the control level of the ICS network architecture and ways to collect it in real-time (Eden et al., 2016; van der Knijff, 2014).

Source	Data	Ways to collect in real-time
Network	Sensor measurements of operating variables in $\mathcal{O} = \{o_1, \ldots, o_m\}$	Network taps and widely available tools like *Wireshark* and *TCPdump*
PLC's	Memory values, states, ladder logic, inputs/outputs (we aggregate this data and denote it as $Config[PLC_i]$ for a given PLC_i)	The proprietary software of the PLC Janicke et al. (2015) and PLC data acquisition tools (e.g., Yau and Chow, 2015)

potential failure occurs, and, as such, it can be a suitable trigger for proactive data collection. The second warning level suggests the need for immediate safety measures, such as engaging a backup controller (Chen and Sankaranarayanan, 2017). In this paper, we focus on the case where a low-criticality warning is raised and we consider it as a suitable trigger for proactive, real-time data collection activities.

5.3. Identifying relevant data (P2)

Of the large volume of data generated by an ICS, we would like to only collect data that is likely to be lost due to damage caused by a potential stealthy attack. We assume that this data is more likely to constitute potential evidence of such an attack, and as such is *relevant* (Pasquale et al., 2018) to our proactive data collection activities. In this paper, since we are concerned with attacks that affect the physical process, we only focus on data originating from control devices that are at the lowest level of the ICS network architecture (Eden et al., 2016; van der Knijff, 2014). Namely, we focus on PLC's and on sensor measurements. We denote by $\mathcal{O} = \{o_1, \ldots, o_m\}$ the set of the system's output variables and by $PLC = \{PLC_1, \ldots, PLC_{n_{plc}}\}$ the set of PLC's in the system. Table 1 summarizes the kind of data we can collect from these devices. For each PLC_i, we denote by $Config[PLC_i]$ the data we can collect from PLC_i.

Unlike data found in process historians, the data that we propose to collect in real-time (Table 1) is assumed to be acquired using dedicated forensic tools, which can help explain how an attack may have happened (Ahmed et al., 2017; Janicke et al., 2015; Yau and Chow, 2015). Although some sensor measurements that we propose to collect can be collected by a process historian, other sensor measurements are collected using network forensic tools that include information, for instance, about source and target IP addresses. This information may be helpful in understanding how a potential attack occurred — e.g., tracing IP addresses to an infected host machine.

In a geographically distributed ICS, the layout of process equipment is designed by taking into account safety properties. Safety-critical process equipment, such as reactors, have an associated *explosion energy*, a measure of the energy that would be released should a safety failure take place within the equipment. This measure determines safe distances that separate different process equipment such that physical damage in one piece of equipment has a low probability of affecting another safety-critical physical process. These energy measures and associated distances are then used to design the layout of the process control system. The result is a partition of the land area available for the plant into different sub-areas where safety-critical equipment should be placed, as shown in Fig. 2 (Quiroz-Pérez et al., 2021). In the process industry this design-time activity is referred to as safe Process Plant Layout (PPL) (Quiroz-Pérez et al., 2021). To reduce the chance of damage spreading from one area into another, protection devices can be installed (Penteado and Ciric, 1996). Note that such layout design does not only account for damage that could result from an explosion, but also from other incidents like chemical spills and fire.

In the TEP example (Fig. 2), we assume that the equipment running the safety-critical processes is geographically distributed according to a safe PPL design (Quiroz-Pérez et al., 2021). In this case, if damage occurs in the reactor for instance, then we assume that the damage will be restricted to the control devices located in the vicinity of the reactor (e.g., a PLC implementing the reactor's temperature control), and will not extend to devices elsewhere. In this sense, we can define the set of process sub-areas as a partition of the set of control devices *PLC*. Namely, let[2] $\mathcal{A} = \{A_1, A_2, \ldots, A_{n_A}\} \subset 2^{PLC}$ be the set of the ICS sub-areas, then $A_i \neq \emptyset,\ A_i \cap A_j = \emptyset,\ \bigcup_i A_i = PLC;\ \forall i, j \in \{1, 2, \ldots, n_A\},\ i \neq j$.

In such ICS, safety constraints are usually expressed as linear combinations of output variables. Namely, let $\phi := \Sigma_{i=1}^{m} a_i o_i \diamond b$ be a safety constraint, with $a_i, b \in \mathbb{R}$, $\diamond \in \{\geq, \leq\}$, and $o_i \in \mathcal{O}$, and let Φ be the set of safety constraints in a given system. Damage in a sub-area A_i can happen due to a violated safety constraint $\phi_i \in \Phi$. We denote by $\text{safe}(\phi_i) \in \mathcal{A}$ the plant area or the subset of control devices that will be affected by the violation of the safety constraint ϕ_i, and by $\text{var}(\phi_i)$ the subset of output variables associated with ϕ_i. This is normally determined by process engineers at the PPL phase (Quiroz-Pérez et al., 2021). Table 2 lists the main safety constraints of the TEP and associates each constraint with the plant area (Fig. 2) where the corresponding safety-critical process is located. For example, constraint ϕ_1 sets an upper limit on the pressure inside the reactor. Since the reactor is located in area A_1, a violation of this constraint could lead to damaging the equipment in A_1; i.e., $\text{safe}(\phi_1) = A_1$.

Recall that the proposed data collection activity is initiated by a low-criticality warning at a time instant k from predictive safety checks that return a subset $\Phi_v(k) \subseteq \Phi$ of safety constraints that may be violated in the future (Azzam et al., 2021; 2022). Consequently, given this subset of safety constraints, we can use Table 2 to determine a set of relevant device and network data (Table 1) at time k, as follows:

$$Config[PLC]_{rel}(k) = \{Config[PLC_i] \mid PLC_i \in \text{safe}(\phi) \forall\ \phi \in \Phi_v(k)\} \tag{1}$$

$$\mathcal{O}_{rel}(k) = \{o \in \text{var}(\phi)\ \forall\ \phi \in \Phi_v(k)\} \tag{2}$$

Consequently, the set of relevant data that we are in a position to collect at a time k is $D_{rel}(k) = Config[PLC]_{rel}(k) \cup \mathcal{O}_{rel}(k)$.

5.4. Deciding which data should be collected (P3)

Real-time data collection activities are generally associated with a certain cost. Additionally, our original predictive safety checks are uncertain as they return a probability of damage, as given by the suspicion metric $s(k)$ (Azzam et al., 2021) at a time instant k. From this observation, it is necessary to decide whether to collect each data in $D_{rel}(k)$ based on a trade-off between the *expected* impact of the attack and the collection cost associated with each data in $D_{rel}(k)$. Namely, we seek to select a subset of data to collect,

[2] Given a set A, 2^A denotes its power set.

Table 2
Safety constraints of the TE case study and affected plant area(s) (Downs and Vogel, 1993).

Constraint label	Output variable (s) [var(ϕ)]	Constraint formula	Affected plant area [safe(ϕ)]
ϕ_1	Reactor Pressure,o_1	$o_1 \leq 2895$ kPa	A_1
ϕ_2	Reactor Temperature o_2	$o_2 \leq 150$ °C	A_1
ϕ_3	Reactor Level o_3	$o_3 \leq 21.3$ m^3	A_1
ϕ_4	Reactor Level o_3	$o_3 \geq 11.8$ m^3	A_1
ϕ_5	Product Separator Level o_4	$o_4 \leq 9.0$ m^3	A_3
ϕ_6	Product Separator Level o_4	$o_4 \geq 9.0$ m^3	A_3
ϕ_7	Stripper Base Level o_5	$o_5 \leq 6.6$ m^3	A_4
ϕ_8	Stripper Base Level o_5	$o_5 \geq 3.5$ m^3	A_4

$D_{col}(k) \subseteq D_{rel}(k)$. To achieve this trade-off, we propose a decision-theoretic framework based on the Value of Information (VoI) analysis. Our main assumption in the development of this framework is that each of the data identified in $D_{rel}(k)$ is equally likely to constitute potential evidence. For example, if $D_{rel}(k)$ consists of a pressure sensor measurement and a configuration of the PLC implementing the pressure controller, then we assume that it is equally likely that the attacker performed their attack by either changing the code of the PLC (Section 4) or intercepting and modifying the pressure sensor measurement. Thus, the only trade-off we are concerned with is the one between the expected impact of the attack and the cost of data collection.

VoI representation We assume that the ICS, represented as $\Pi(\theta)$, operates in nominal conditions under a certain performance measured by a revenue $W(\theta)$ that depends on some system parameters grouped in θ. For example, the TEP's performance is usually measured by the revenue resulting from the amount of chemical product produced (Downs and Vogel, 1993; Ricker, 1996). The operational cost of $\Pi(\theta)$ is denoted by $C(\theta)$, and can be estimated based on a variety of parameters; such as the amount of reactants consumed in the TEP case. In this paper, we restrict this cost to the cost of data collection, i.e., collecting the data in the set D_{rel}.

Under a given parameter setting θ, the value (profit) of $\Pi(\theta)$ is given by $V(\theta) = W(\theta) - C(\theta)$. However, due to uncertainties associated with parameters θ, we must use an *expected* parameter estimate to compute $V(\theta)$. Consequently, we consider the expected value (profit), denoted by $\mathbb{E}[V(\theta)]$, under expected parameters $\mathbb{E}[\theta]$.

Suppose that we predict that a potential stealthy attack, may bring the system into a new operating mode with $E[V(\tilde{\theta})] < E[V(\theta)]$, by causing damage to one or more of the plant areas A_i (i.e., we are presented with the trigger described in Section 5.2). Then we can ask whether it is worthwhile to collect certain relevant data (as identified in $D_{rel}(k)$) that may allow us to reveal a breach like the one illustrated in Section 4. Namely, we may want to collect a given $\delta_i \in D_{rel}(k)$ only if its collection cost does not exceed the potential reduction in the value or the revenue of the system's operation (i.e., attack impact) if the predicted stealthy attack is successful. The change in this expected value is given by:

$$\begin{aligned}\Delta_V(\tilde{\theta}, \theta) = \mathbb{E}[V(\tilde{\theta})] - \mathbb{E}[V(\theta)] = \mathbb{E}[W(\tilde{\theta}) - C(\tilde{\theta})] \\ -\mathbb{E}[W(\theta) - C(\theta)]\end{aligned} \tag{3}$$

As a simplifying assumption, we can take the cost of data collection under nominal conditions $C[\theta]$ as 0. On the one hand, if we choose not to collect any data and subsequently not attempt to prevent the damage from the breach, then $C[\tilde{\theta}] = 0$ and in this case $|\Delta_V(\tilde{\theta}, \theta)| = |\mathbb{E}[W(\tilde{\theta}) - W(\theta)]| = \mathbb{E}[|\Delta_W|]$. On the other hand, if we choose to collect data that can help us identify the location of the breach and prevent damage from happening, then we can assume that $W(\tilde{\theta}) = W(\theta)$ and $|\Delta_V(\tilde{\theta}, \theta)| = \mathbb{E}[C(\tilde{\theta})]$. In other words, the collection of data that may reveal a suspected breach is worthwhile only if the expected cost of collection does not exceed the expected reduction in the operating revenue (performance), $\mathbb{E}[|\Delta_W|]$, i.e., the expected impact of the attack.

Example Consider the TEP system generating a value of 0.8 under nominal operation. Our online safety monitor predicts at a time instant k that a potential breach may be able to cause damage to two out of the five plant areas in $\mathcal{A}$ with a probability (given by the suspicion metric) of $s(k) = P(\text{damage}) = 0.4$. If this happens, then the new revenue of operation is 0.48. As such, the new expected revenue of operation is given by:

$$\begin{aligned}\mathbb{E}[\tilde{W}] &= P(\text{damage})(W|\text{damage}) + P(\neg\text{damage})(W|\neg\text{damage}) \\ &= 0.4 \times 0.48 + 0.6 \times 0.8 = 0.612\end{aligned} \tag{4}$$

Hence, the expected reduction in revenue of operation (i.e., attack impact) is $\mathbb{E}[|\Delta_W|] = 0.8 - 0.612 = 0.188$. This is the maximum we will "pay" to identify the location of the suspected breach, i.e., collect each of the data in $D_{rel}(k)$.

Cost of data collection In the set $D_{rel}(k)$ that we identified in Section 5.3, the collection of each of the data $\delta_i \in D_{rel}(k)$ is associated with a certain cost $C(\delta_i)$. This measure of collection cost can be estimated, for example, based on the capabilities of the existing data acquisition tools. Namely, if the collected data is to be stored in some secure server, one could set a limit on the total size of data that we can collect at a given time instant k. Then, we can define a collection cost $C(\delta_i)$ as the ratio of the space occupied by δ_i to the size limit for data collection at a time k. In a similar fashion, this cost measure can account for bandwidth limitations and data transmission delays that may arise in large-scale geographically-distributed ICS. Regardless of the exact definition of $C(\delta_i)$, we propose that this measure be a dimensionless number in the same range as the measure of revenue of operation ($W(\theta)$), so we can soundly perform the trade-off as illustrated in the previous example.

Going back to the previous example, assume that the identified relevant data consists of the pressure sensor measurement o_1 and the ladder logic of the PLC implementing the pressure controller, $Config[PLC_1]$. Since a single sensor measurement may occupy significantly less space that a PLC ladder logic configuration, we assume that the cost of collecting o_1 is $C(o_1) = 0.02$ and that of $Config[PLC_1]$ is $C[Config[PLC_1]] = 0.2$. Since the maximum we will pay to identify the location of the suspected breach is 0.188, then we can conclude that o_1 is worth collecting while $Config[PLC_1]$ is not.

5.5. Real-time proactive data collection

We implement the three steps of our approach proposed to collect relevant data in Algorithm 1. The execution of Algorithm 1 at a time instant k is triggered by a low-criticality warning from the online safety monitor (Azzam et al., 2021; 2022) which returns the suspicion metric $s(k)$, i.e., probability of damage taking place, in addition to the potentially violated constraints $\Phi_v(k)$ (in the future). We first use the safe PPL and Table 2 to identify data that

Algorithm 1 Proactive real-time collection of relevant data.

INPUTS: A subset of potentially violated constraints $\Phi_v(k)$; the suspicion metric $s(k) = P(\text{damage})$
PARAMETERS: collection cost of potential data types $C(\delta_i)$, ICS operating value under nominal conditions $(W|\neg\text{damage})$
OUTPUTS: $D_{\text{col}}(k)$ (a set of data to collect)

1▸ **for all** $\phi \in \Phi_v(k)$ **do**
2▸ $Config[PLC]_{rel}](k) \leftarrow Config[PLC]_{rel}](k) \cup \text{safe}(\phi)$
3▸ $\mathcal{O}(k)_{rel} \leftarrow \mathcal{O}_{rel}(k) \cup \text{var}(\phi)$
4▸ **end for**
5▸ $D_{rel}(k) \leftarrow \mathcal{O}_{rel}(k) \cup Config[PLC]_{rel}](k)$
6▸ $\mathbb{E}[\tilde{W}] \leftarrow s(k)(W|\text{damage}) + (1 - s(k))(W|\neg\text{damage})$
7▸ $\text{MaxCollectionCost} \leftarrow (W|\neg B) - \mathbb{E}[\tilde{W}]$
8▸ **for all** $\delta_{rel} \in D_{rel}(k)$ **do**
9▸ **if** $C(\delta_{rel}) <$ MaxCollectionCost **then**
10▸ $D_{col}(k) \leftarrow D_{col}(k) \cup \{\delta_{rel}\}$
11▸ **end if**
12▸ **end for**

is more likely to be lost due to the predicted damage. We then perform the VoI analysis on the identified relevant data in $D_{rel}(k)$. First, we estimate the expected reduction in revenue (impact) from the predicted attack, as illustrated in the previous example. We use this computed impact as the maximum we will pay to collect a certain $\delta_{rel} \in D_{rel}(k)$ and subsequently make our collection decision given an associated $C(\delta_{rel})$.

Note that the estimation of the revenue if damage happens, $(W|\text{damage})$, can be performed using the number of areas A_i that are predicted to be damaged by the online safety monitor. In our numerical example, the safety monitor predicted that two out of five areas in $\mathcal{A}$ may be damaged. As such, we can estimate a $2/5 = 40\%$ reduction in the original revenue under nominal operation. Note that a better estimate can be obtained by incorporating some process engineering knowledge about the contribution of each plant area to the revenue; however, such considerations are beyond the scope of the present work.

6. Evaluation

This section discusses our evaluation strategy, describes the testbed that we use in the evaluation, and presents the results.

6.1. Overview

Our evaluation does not include a discussion of the trigger for data collection (P1), which is based on previous work (Azzam et al., 2021; 2022). In this paper, we evaluate the relevance of the collected data (P2), and assess the reduction in control performance overhead obtained using our decision-theoretic framework (P3). We then demonstrate a use case of our approach using an attack scenario. This evaluation is consistent with previous work on FR (Alrajeh et al., 2017). While Alrajeh et al. (2017) employed existing datasets to perform their evaluation, we resort in our work to simulations of a virtual realistic testbed. The reason for using simulations stems from the lack of datasets that we can use to evaluate our work. To the best of our knowledge, datasets on ICS security (e.g., Beaver et al., 2013) are not constructed to proactively collect data that can be relevant to identify how a stealthy attack occurred. Existing datasets are instead aimed to train and test statistical models for attack detection. Thus, they are not suitable to evaluate our approach. Data collected by process historians, as illustrated in Section 4, may not explain how an attack occurred. Thus, we avoid a comparison of our approach with process historians.

6.2. Testbed description

We employed a simulation of the TEP implemented in MATLAB/Simulink by Bathelt et al. (2015) based on the control architecture designed by Ricker (1996). We augmented the simulation with an implementation of the algorithm proposed by Azzam et al. (2021), and the addition of Algorithm 1.[3] The simulation also includes blocks to simulate networked behaviour and data transmission delays according to the setup given in Azzam et al. (2021). To simulate the process of data collection, we augment the controllers in the TEP simulation with logging capabilities. Namely, when Algorithm 1 identifies the set $D_{col}(k)$, the corresponding controllers generate a log in a `json` format. In particular, we assume that each of the controllers designed by Ricker (1996) are implemented on a PLC. As such, we simulate the collection of data from these devices by adopting a log structure that mimics the logging of PLC ladder logic configurations (e.g., Yau et al., 2018). Namely, each log entry consists of a timestamp, sensor measurements received, and details about the controller configuration, in addition to the output of the controller (actuation signal). Since each controller in the architecture proposed by Ricker (1996) is a discrete-time Proportional-Integral (PI) controller, the controller configuration is represented by the values of the coefficients of the proportional and integral terms in addition to the sampling time. For sensor measurements, the corresponding sensors simply generate a `json` log entry that includes the timestamp, the value, and the name of the sensor measurement (Table 2). We performed all the simulations described in this paper on a machine with an Intel i7-9750H CPU clocked at 2.6 GHz with 16 GB of memory.

6.3. Relevance and overhead reduction

To assess the relevance of collected data and the reduction in performance overhead resulting from data collection, we adopt a large number of randomised (Monte Carlo) simulations. In each simulation, we randomise the operating levels of the TEP (i.e., setpoints for pressure, temperature, reactor level, etc.), and we choose attacked sensors randomly. For relevance, we check in each simulation whether any data that was lost to damage by the attack was missed by the set identified in step (P2) ($D_{col}(k)$).

The decision-theoretic framework proposed in step (P3) already ensures, by design, that the collection of data is enabled only when the expected reduction in system performance exceeds the cost of data collection. Thus, we do not compare the revenue before and after data collection. Instead, we showcase how, as a result of the decision-theoretic framework, there is a limited effect on the system's performance. We compare the average execution time of controllers under forensic logging enabled by Algorithm 1, with the case where forensic logging is enabled at all times from all controllers. We also present the average reduction in terms of the number of log entries between the two scenarios. In addition, and as a reference, we present the average execution time of controllers in the case where no logging is enabled. Although our logging mechanism is a simplified version of data acquisition tools, the comparison of average controller execution times between the aforementioned scenarios serves to give an idea of the potential performance improvement brought by our approach with actual data acquisition tools. For each of the testing scenarios previously described, we ran a total of 1000 simulations. Results are shown in Table 3.

Discussion The low percentage of data missed by the proposed algorithm as shown in Table 3 demonstrates the capability of our

[3] These implementations and the scripts used to perform the evaluation can be found at https://github.com/ul-res/ics-forensic-readiness.

Table 3
Simulation results.

Test	Result
Relevance (percentage of cases where data at risk of being damaged is missed)	5.8%
Average controllers' execution time with Algorithm 1	1.5×10^{-3} s
Average controllers' execution time with all data collection enabled	2.3×10^{-3} s
Average reduction in log size when using Algorithm 1	30.8%

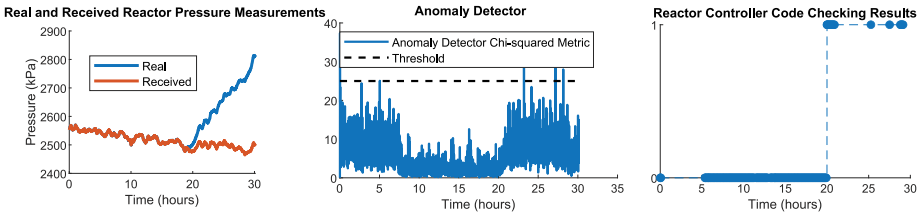

Fig. 3. Real and received sensor measurements, anomaly detector metric, and results from the real-time PLC code checking applied with the proposed approach (1 = detected code modification, 0 = otherwise).

approach to identify relevant data (P2), thus, limiting the loss of potential evidence due to damage by a stealthy attack. Additionally, the use of Algorithm 1 with the proposed decision-theoretic framework (P3) results in reducing the average controllers' execution time by around 35%, compared with the case where logging was performed from all controllers at all times. This is consistent with the 31% reduction in the size of logs due to Algorithm 1. These results show that the proposed decision-theoretic framework enables the collection of data in a way that minimises the interference with control operations.

6.4. Use case: supporting PLC log-based live forensics

To demonstrate a potential use-case of our approach, we consider tools that can perform live forensics in ICS, i.e., investigating potential attacks while the system is running. Due to the safety-criticality of ICS and the difficulty associated with shutting them down, methods for live forensics activities (Al-Sharif et al., 2018) have been proposed to avoid such difficulties. One important requirement for such methods is that they need to minimise interference with the safety-critical control operations (Eden et al., 2016). In the previous section, we showed how our approach reduces the number of PLC logs that need to be collected, hence reducing the effect of logging mechanisms on the controllers' execution time. In this section, we further show how our approach can be used to improve the performance of live forensics operations, due to the fact that we only consider relevant data and we account for the system's performance while guiding logging activities.

A class of ICS live forensics methods (Yau and Chow, 2015) consists of investigating PLC logs to check for any malicious modification to their code. To showcase the increase in performance of such methods, we revisit the Stuxnet-inspired attack scenario introduced in Section 4. In the second phase of the attack, the attacker modifies the function codes in the PLC controlling the reactor pressure in a way that drives the reactor slowly beyond safe operating regions. We run a simulation of such an attack by introducing malicious code to the controller handling the reactor pressure. We augment the implementation of the Algorithm 1 and the logging mechanism described previously with an implementation of the technique proposed by Yau and Chow (2015) based on the log structure that we use in our simulations. Whenever controller logs are generated, we compare the logged control output with the one expected by the originally designed control law (in Ricker, 1996). If the two values differ substantially, i.e., more than a pre-defined threshold, the code checking detects the presence of a code modification (Fig. 3).

Table 4
PLC code checking performance.

Test	Average real-time execution (s)
With Algorithm 1	0.0013
Without Algorithm 1	0.0277

Fig. 3 shows a plot of the received and actual (real) pressure measurements from the reactor under an integrity attack on pressure sensor, along with the anomaly detector's detection metric. In addition, we show a plot of the results from the code modification checks performed to the pressure controller logs collected according to our approach. Namely, we plot the results from the code checking vs. the timestamps of the collected logs. We also analyse the performance of PLC code checking under two cases: 1) the checking is performed on logs generated according to Algorithm 1 and 2) the checking has to consider logs from all controllers at each time instant (Table 4). The results were averaged over a $\approx$ 30-h simulation, equivalent to around 60,000 real-time checks given the system's 1.8 s sampling time.

Discussion Fig. 3 shows that our approach enabled the collection of logs from the reactor's pressure controller, where code checking was able to detect modification of the PLC code well before any alarm was raised by the existing anomaly detector. While our approach only collected logs from the pressure controller at the start of the attack, it enabled the detection of code modifications much earlier than the existing anomaly detector. In the case where logs are needed for the duration of attacks, Algorithm 1 can be modified such that it can be overridden to ensure the collection of logs as long as the code checking returns with the result that the code is indeed modified.

The benefit of performing PLC code checking under the proposed approach is highlighted with the results shown in Table 4. Under this approach, the average real-time execution time of code checking was reduced by more than an order of magnitude. In ICS, live forensics is used more often than post-mortem forensics (Eden et al., 2016) to avoid the high-costs of shutting down the process. The performance enhancement brought about by the proposed approach has the potential of reducing the risk of interfering with the real-time performance constraints of safety-critical ICS when live forensics is performed, especially considering that devices such as PLC's feature low computational resources.

Based on the results obtained in this section, it can be expected that our approach may also improve the performance of other log-based attack detection techniques. As our approach specifies when

and which data is worth collecting at a given time, this reduces the amount of logs that need to be processed by such techniques. For example, the technique presented by Hussain et al. (2019) relies on converting logs from different ICS components into a petri net representation in order to detect anomalies based on existing process knowledge. In the future, it may be worth investigating how the reduction in log sizes can affect the performance of such conversion, and whether it can help increase the precision of the anomaly detection technique. Our approach could also use machine learning to improve the selection of logs for this purpose.

6.4.1. Potential limitations

The use case presented in this section has highlighted a potential limitation with our approach: data collection may be triggered for a long time even if no attack was taking place, as shown in Fig. 3. However, live forensics tools may serve as a heuristic to determine whether proactively collected data is to be preserved or discarded. For instance, in our use case scenario, the PLC logs collected when no attack was taking place as shown by the code checking could be discarded in order to save storage resources. Future work could look into more ways in which data collected due to our approach can be handled in order to optimize storage capabilities.

Another potential limitation with our approach is that it focuses on a specific type of attacks on ICS. While the model we considered is widely studied, further work would be needed to consider a variety of other attacks. The approach presented in this paper serves as a framework to consider data collection for future investigations of other types of attacks. Namely, one can have a certain metric reflecting a "probability of harm", similar to the suspicion metric, for other attack types. Our framework for relevant data identification and selection can then be used based on the new metric.

7. Conclusion

In this paper, we propose an approach to engineer Forensic-Ready safety-critical, geographically distributed ICS presented with the threat of stealthy attacks targeting physical processes. We designed an approach to proactively collect relevant data before they are lost due to damage caused by a stealthy attack. In the absence of potential alarms about stealthy attacks from anomaly detectors, our approach considers an alternative trigger for data collection based on predictive safety checks. Then, using the plant layout and the constraints that are predicted to be violated, we identify data that are most likely to be lost due to damage. We also propose a decision-theoretic framework that enables the collection of data only when collection cost does not exceed the expected reduction in the system's performance due to the predicted attack.

Our evaluation of the proposed approach has shown that it ensures the relevance of collected data and incurs a limited control performance overhead. Furthermore, we demonstrated the advantage of such an approach in improving the efficiency of existing live forensic log analysis techniques. In future work we will incorporate process engineering knowledge to better estimate the potential impact of a predicted attack in our decision-theoretic framework. We will also evaluate our approach on a live testbed using data acquisition tools, and investigate how the approach can help improve the performance and efficiency of attack detection techniques.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Mazen Azzam: Data curation, Formal analysis, Investigation, Software, Writing – original draft. **Liliana Pasquale:** Supervision, Validation, Writing – review & editing. **Gregory Provan:** Formal analysis, Supervision, Validation. **Bashar Nuseibeh:** Funding acquisition, Project administration, Supervision.

Data Availability

Data will be made available on request.

Acknowledgement

This work was supported by Science Foundation Ireland grants 13/RC/2094 P2 and 16/RC/3918 and the Engineering and Physical Sciences Research Council (EPSRC) grant EP/R013144/1.

References

Ab Rahman, N.H., Glisson, W.B., Yang, Y., Choo, K.K.R., 2016. Forensic-by-design framework for cyber-physical cloud systems. IEEE Cloud Comput. 3 (1), 50–59.
Ahmed, I., Obermeier, S., Sudhakaran, S., Roussev, V., 2017. Programmable logic controller forensics. IEEE Secur. Privacy 15 (6), 18–24.
Al-Sharif, Z.A., Al-Saleh, M.I., Alawneh, L.M., Jararweh, Y.I., Gupta, B., 2018. Live forensics of software attacks on cyber-physical systems. Future Gener. Comput. Syst. 108, 1217–1229.
Albright, D., Brannan, P., Walrond, C., 2011. Stuxnet malware and Natanz: update of ISIS December 22, 2010 report. Inst. Sci. Int. Secur. 15.
Alrajeh, D., Pasquale, L., Nuseibeh, B., 2017. On evidence preservation requirements for forensic-ready systems. In: Proc. of the 11th Joint Meeting on Foundations of Software Engineering. ACM, pp. 559–569.
Altschaffel, R., Hildebrandt, M., Kiltz, S., Dittmann, J., 2019. Digital forensics in industrial control systems. In: International Conference on Computer Safety, Reliability, and Security. Springer, pp. 128–136.
Awad, R.A., Beztchi, S., Smith, J.M., Lyles, B., Prowell, S., 2018. Tools, techniques, and methodologies: a survey of digital forensics for SCADA systems. In: Proc. of the 4th Annual Industrial Control System Security Workshop. ACM, pp. 1–8.
Azzam, M., Pasquale, L., Provan, G., Nuseibeh, B., 2021. Grounds for suspicion: physics-based early warnings for stealthy attacks on industrial control systems. IEEE Trans. Dependable Secure Comput. 1. doi:10.1109/TDSC.2021.3113989.
Azzam, M., Pasquale, L., Provan, G., Nuseibeh, B., 2022. Efficient predictive monitoring of linear time-invariant systems under stealthy attacks. IEEE Trans. Control Syst. Technol. 1–13. doi:10.1109/TCST.2022.3196809.
Bathelt, A., Ricker, N.L., Jelali, M., 2015. Revision of the Tennessee Eastman process model. IFAC-PapersOnLine 48 (8), 309–314.
Beaver, J.M., Borges-Hink, R.C., Buckner, M.A., 2013. An evaluation of machine learning methods to detect malicious SCADA communications. In: Proc of the 12th International Conference on Machine Learning and Applications. IEEE, pp. 54–59.
Carrier, B., Spafford, E., 2004. An event-based digital forensic investigation framework. Digit. Invest..
Chan, C.F., Chow, K.P., Yiu, S.M., Yau, K., 2018. Enhancing the security and forensic capabilities of programmable logic controllers. In: IFIP International Conference on Digital Forensics. Springer, pp. 351–367.
Chen, X., Sankaranarayanan, S., 2017. Model predictive real-time monitoring of linear systems. In: Proc. of the Real-Time Systems Symposium. IEEE, pp. 297–306.
Downs, J.J., Vogel, E.F., 1993. A plant-wide industrial process control problem. Comput. Chem. Eng. 17 (3), 245–255.
Eden, P., Blyth, A., Burnap, P., Cherdantseva, Y., Jones, K., Soulsby, H., Stoddart, K., 2016. Forensic readiness for SCADA/ICS incident response. In: Proceedings of the 4th International Symposium for ICS and SCADA Cyber Security Research. BCS Learning and Development Ltd., pp. 1–9.
Elhoseny, M., Abbas, H., Hassanien, A.E., Muhammad, K., Sangaiah, A.K., 2017. Secure automated forensic investigation for sustainable critical infrastructures compliant with green computing requirements. IEEE Trans. Sustain. Comput. 5 (2), 174–191.
Giraldo, J., Urbina, D., Cardenas, A., Valente, J., Faisal, M., Ruths, J., Tippenhauer, N.O., Sandberg, H., Candell, R., 2018. A survey of physics-based attack detection in cyber-physical systems. ACM Comput. Surv. (CSUR) 51 (4), 76.
Griffioen, P., Weerakkody, S., Sinopoli, B., Ozel, O., Mo, Y., 2019. A tutorial on detecting security attacks on cyber-physical systems. In: 2019 18th European Control Conference (ECC). IEEE, pp. 979–984.
Grispos, G., Glisson, W.B., Choo, K.K.R., 2017. Medical cyber-physical systems development: a forensics-driven approach. In: 2017 IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE). IEEE, pp. 108–113.
Hadžiosmanović, D., Bolzoni, D., Hartel, P.H., 2012. A log mining approach for process monitoring in SCADA. Int. J. Inf. Secur. 11 (4), 231–251.
Hussain, M., Foo, E., Suriadi, S., 2019. An improved industrial control system device logs processing method for process-based anomaly detection. In: 2019

International Conference on Frontiers of Information Technology (FIT). IEEE, pp. 150–1505.
Janicke, H., Nicholson, A., Webber, S., Cau, A., 2015. Runtime-monitoring for industrial control systems. Electronics 4 (4), 995–1017.
Kebande, V.R., Choo, K.K.R., 2022. Finite state machine for cloud forensic readiness as a service (CFRAAS) events. Secur. Privacy 5 (1), e182.
van der Knijff, R.M., 2014. Control systems/SCADA forensics, what's the difference? Digit. Invest. 11 (3), 160–174.
Kwon, C., Hwang, I., 2018. Reachability analysis for safety assurance of cyber-physical systems against cyber attacks. IEEE Trans. Autom. Control 63 (7), 2272–2279.
Lee R., Assante M., Conway T.. SANS ICS Defense Use Case (DUC): ICS CP/PE case study paper-German Steel Mill Cyber Attack. SANS ICS2014.
Mishra, S., 2019. Forensic Investigation Framework for Complex Cyber Attack on Cyber Physical System by Using Goals/Sub-Goals of an Attack and Epidemics of Malware in a System. Springer, pp. 491–504.
Mohamed, N., Al-Jaroodi, J., Jawhar, I., 2020. Cyber-physical systems forensics: today and tomorrow. J. Sens. Actuator Netw. 9 (3), 37.
Murguia, C., Ruths, J., 2018. On reachable sets of hidden CPS sensor attacks. In: 2018 Annual American Control Conference (ACC). IEEE, pp. 178–184.
Myers, D., Radke, K., Suriadi, S., Foo, E., 2017. Process discovery for industrial control system cyber attack detection. In: IFIP International Conference on ICT Systems Security and Privacy Protection. Springer, pp. 61–75.
Pasquale, L., Alrajeh, D., Peersman, C., Tun, T., Nuseibeh, B., Rashid, A., 2018. Towards forensic-ready software systems. In: Proc. of the 40th Int. Conf. on Software Engineering: New Ideas and Emerging Results. IEEE, pp. 9–12.
Pasqualetti, F., Dörfler, F., Bullo, F., 2013. Attack detection and identification in cyber-physical systems. IEEE Trans. Autom. Control 58 (11), 2715–2729.
Penteado, F.D., Ciric, A.R., 1996. An MINLP approach for safe process plant layout. Ind. Eng. Chem. Res. 35 (4), 1354–1361.
Qasim, S.A., Smith, J.M., Ahmed, I., 2020. Control logic forensics framework using built-in decompiler of engineering software in industrial control systems. Forensic Sci. Int. 33, 301013.
Quiroz-Pérez, E., de Lira-Flores, J.A., Gutiérrez-Antonio, C., Vazquez-Román, R., 2021. A new multiple-risk map approach to solve process plant layout considering safety and economic aspects. J. Loss Prev. Process. Ind. 72, 104524.
Ricker, N.L., 1996. Decentralized control of the Tennessee Eastman challenge process. J. Process Control 6 (4), 205–221.
Rowlingson, R., et al., 2004. A ten step process for forensic readiness. Int. J. Digit. Evid. 2 (3), 1–28.
Taveras, P., 2013. SCADA live forensics: real time data acquisition process to detect, prevent or evaluate critical situations. Eur. Sci. J. 9 (21), 253–262.
Urbina, D.I., Giraldo, J.A., Cardenas, A.A., Tippenhauer, N.O., Valente, J., Faisal, M., Ruths, J., Candell, R., Sandberg, H., 2016. Limiting the impact of stealthy attacks on industrial control systems. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. ACM, pp. 1092–1105.
Van Vliet, P., Kechadi, M.T., Le-Khac, N.A., 2015. Forensics in Industrial Control System: A Case Study. Springer, pp. 147–156.
Yau, K., Chow, K.P., 2015. PLC forensics based on control program logic change detection. J. Digit. Forensics, Secur. Law 10 (4), 5.
Yau, K., Chow, K.P., Yiu, S.M., 2018. A forensic logging system for siemens programmable logic controllers. In: IFIP International Conference on Digital Forensics. Springer, pp. 331–349.
Zainudin, N.M., Hasbullah, N.A., Wook, M., Ramli, S., Razali, N.A.M., 2022. Digital forensic readiness for cyber security practitioners: an integrated model. J. Posit. School Psychol. 6 (3), 8423–8433.

Mazen Azzam received his Ph.D. degree from the University of Limerick (Ireland), while working within the Lero - the Irish Software Research Centre and the CONFIRM smart manufacturing research centre. His research interests include control engineering, security, and digital forensics for Cyber-Physical Systems and Industrial Control Systems in particular.

Liliana Pasquale received her Ph.D. degree from Politecnico di Milano (Italy), in 2011. She is assistant professor at University College Dublin (Ireland) and a funded investigator at Lero - the Irish Software Research Centre. Her research interests include requirements engineering and adaptive systems, with particular focus on security, privacy, and digital forensics. She has served in the Program and Organizing Committee of prestigious software engineering conferences, such as ICSE, FSE, ASE, RE. She is also part of the review committee of the IEEE TSE journal and the TOSEM journal.

Gregory Provan is a Professor at the Computer Science Department at University College Cork (UCC), in Cork, Ireland. His research interests include the modelling and analysis of complex systems, in particular modelling for control and diagnostics purposes, and the use of machine learning for modelling and optimization. He is currently conducting research as part of the Insight and LERO Centres as funded by Science Foundation Ireland.

Bashar Nuseibeh is a Professor of Software Engineering at the University of Limerick and Chief Scientist at Lero - The Irish Software Research Centre. He is also a Professor of Computing at the Open University, an Honorary Professor at University College London, and a Visiting Professor at the National Institute of Informatics, Japan. His research interests include software requirements and design, security & privacy, and the engineering of adaptive systems. He is a co-principal investigator in Confirm – the SFI research centre on Smart Manufacturing. He has served as editor-in chief of IEEE Transactions on Software Engineering, ACM Transactions on Autonomous and Adaptive Systems, and the Automated Software Engineering Journal. He is an associate editor of IEEE Security & Privacy magazine. Bashar is a Fellow of the British and Irish Computer Societies, a Fellow of the Institution of Engineering & Technology, and a Member of Academia Europaea and the Royal Irish Academy. More information at http://nuseibeh.com