Designing Safe and Secure Industrial Control Systems: A Tutorial Review

Dimitrios Serpanos
ISI/ATHENA and University of Patras

Howard Shrobe
MIT

Muhammad Taimoor Khan
University of Klagenfurt

Editor's Note:
This tutorial deals with the increasing number of cyber attacks in industrial control system which lead to increasing economical damage. The authors focus on the most relevant topics including how to design such systems in future with the goal of higher safety and security. The reader will first learn the basics like the deployed architectures and system layers after which the discussion turns to design aspects, intrusion detection and prevention followed by a survey of current research trends.

—Jörg Henkel, Karlsruhe Institute of Technology

Cyberattacks have been steadily increasing and their effects have become visible in everyday life. Sensitive information leakage from financial institutions such as Equifax [8], service disruption attacks to popular web services such as the recent attack on GitHub [54], ransomware attacks against individuals and organizations [9], and other attacks hit the news often. In the past decades, we have witnessed increasing attacks on critical infrastructures as well [50]. Power plants, factories, and water processing facilities have experienced successful attacks that have disrupted operations and influenced the life of large population groups. The increasing number of attacks against infrastructure is especially troublesome, considering that it not only influences the lives of large populations in the short term but also it can lead to significant damages to economies and affect the well-being and progress of populations in the long run; furthermore, infrastructural problems, such as loss of power or transportation system disruptions, can lead to loss of life.

Critical infrastructures, today, are run by industrial control systems (ICSs), a class of industrial computers that are interconnected with a wide range of networks, including specialized industrial networks and the Internet. ICSs have emerged in parallel with typical information technology (IT) computing systems and networks with the purpose to implement and manage industrial processes. Since most of these systems were originally isolated and dedicated to special purpose critical applications, their development methods evolved separately and had different goals and priorities. For example, requirements on typical ICSs include continuous operation and real-time response, leading to systems whose software is not easily upgraded or patched as in traditional IT systems. Importantly, although safety has traditionally been a significant requirement for ICS, security has not been a major concern until recently, when the increased

Digital Object Identifier 10.1109/MDAT.2018.2816943
Date of publication: 20 March 2018; date of current version: 21 May 2018.

Copublished by the IEEE CEDA, IEEE CASS, IEEE SSCS, and TTTC

connectivity of the systems as well as the adoption of automated process management in a large-scale exposed ICSs to attacks that had never been considered in the past. Clearly, security has become a major priority in ICSs, since it affects safety as well.

In this paper, we review methods to design safe and secure ICSs. First, we introduce the typical ICS architecture. We describe the relationship between safety and security in the context of ICSs and present threats against them. Considering that ICSs need to be designed with appropriate security functions; and then, they need to be protected during their operation, we review methods and techniques for design as well as for monitoring at runtime.

Industrial control systems

ICSs constitute a class of cyber-physical systems that implement industrial processes. Although they were originally developed and employed in typical industrial environments for industrial control, their use has extended to the control and management of a wide range of processes, from avionics to power grids, from traffic management and transport systems to water management. Today, ICSs are employed for the management and control of most of the critical infrastructure of countries. Although ICS are computing systems, their development, management, and operation differs from traditional IT systems, since they are characterized by different interfaces, they are owned and managed by different engineers, they have strong requirements for continuous operation and realtime, and they employ specialized network protocols. Due to this, they are designated as operational technology (OT) systems.

ICSs typically implement a control loop, as shown in Figure 1, controlling a physical process (also called a plant) that is composed of one or multiple physical devices. In the control loop, sensors take measurements of parameters, they deliver them to a control center that executes the necessary computations, and outputs commands to actuators.

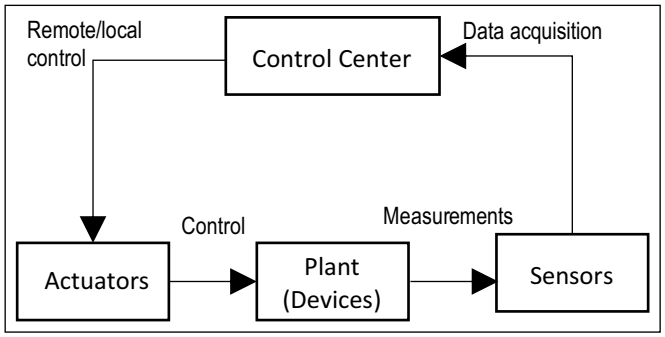

Figure 1. Control loop view.

The control loop, which constitutes the application view of an industrial process, is implemented through a hierarchy of industrial computing systems as shown in Figure 2. A programmable logic controller (PLC) is a computing system that implements two logical processes: 1) it controls autonomously the connected device(s) at the lower level of the hierarchy, taking as input sensor data and controlling actuators and 2) it executes a component of a distributed application that controls the whole plant under the supervision of the supervisory control and data acquisition (SCADA) system, communicating with the SCADA system and, possibly, with other PLCs. Thus, the industrial process, i.e., the application of the ICSs, is designed and implemented as a distributed computing application, being decomposed to communicating computational processes that are mapped to the hierarchy of the ICSs shown in Figure 2, i.e., it is designed as a system of systems [48], [56].

ICSs are safety-critical, since they are employed in application domains ranging from avionics and manufacturing to smart grids and water management. Their failures, whether accidental or intentional, can have catastrophic results, damaging infrastructures, property, and even people. Importantly, the significant effects of the operational disruption of their applications have attracted the attention of actors who target ICSs, launching attacks on them and causing significant operational problems to targets. The recent attacks on the Ukranian electrical grid [79] and the well-known Mirai attack in 2016 [55] are only a few known events that follow the Aurora experiment [77], where an engine was destroyed only by cyber means in a controlled experiment, and Stuxnet [44], the first documented cyberattack on ICSs, which caused a significant setback to the Iranian nuclear program. So, safety and security of ICSs are fundamental properties of emerging systems. Safety has been a major concern and goal of ICSs, but it has focused mostly on accidental failures. However, the emergence of malicious attacks requires a unified approach to safety and security, since malicious actors launch attacks that create conditions for accidents intentionally and without any degree of randomness as traditional models of faults and failures consider.

Emerging ICSs need to be designed with several requirements. Their typical employment

in safety-critical and often real-time applications leads to requirements for resilience and robustness, achieving continuous operation and availability of applications and services, performing the critical operations even under conditions of failures or attacks, and recovering from such incidents. In order to design safe and secure ICSs, we need to address two important aspects: 1) design the system correctly, i.e., satisfying the set requirements for safety and security, including resilience and robustness and 2) to include in the design run-time monitor(s) that will detect attacks and failures, considering that systems cannot be protected from all possible failures and attacks. Correct system design that meets safety and security requirements, i.e., a secure-by-design system, guarantees a correct starting point of system operation, while monitoring enables run-time defense, avoidance of hazardous operation and, potentially, recovery from failure and attack incidents.

Security and safety in industrial control systems

Safety and security are terms that have different meaning to engineers and scientists of different disciplines and backgrounds; often, they are even used interchangeably. Until recently, safety and security were considered independently as different disciplines and with different engineering methods. As a result, it is often unclear what constitutes a safe and/or secure system and what is the relationship between safety and security.

A general definition of safety is provided by Leveson [45] as "freedom from accidents or losses." Independently of the exact definition one may use for safety, there is one common characteristic in definitions of safety: it is application process related and specific. On the other hand, security is typically considered as a computing issue; in general, computer and network security is the set of mechanisms and policies that protect computing systems and networks from data leakage, misuse, alteration, and loss, as well as from operation disruption, service misuse, denial of service (DoS), and theft of resources, whether hardware or software. Clearly, there are several common aspects in safety and security, although the main focus of safety is the physical process, while the focus of security is the computing systems and networks.

In order to address safety and security in ICSs unambiguously, we view ICSs as layered systems,

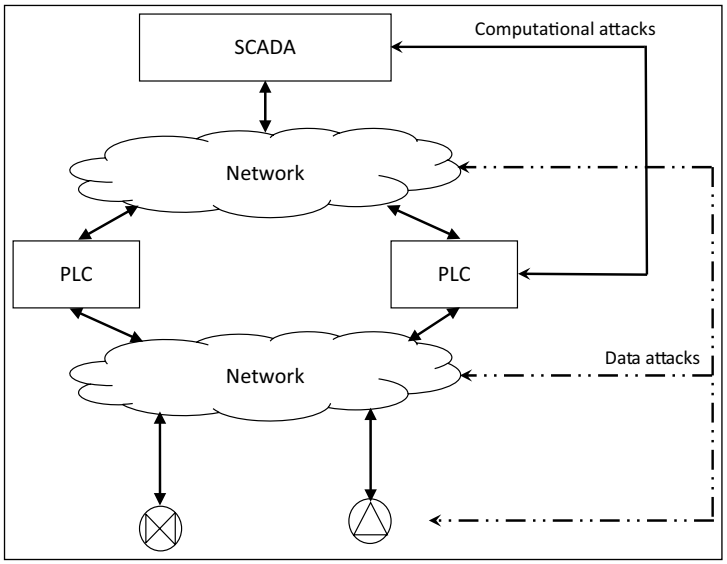

Figure 2. ICS view.

shown in Figure 3, where safety and security requirements are set at different layers. We consider that safety requirements refer to the physical process implemented through the application of an ICS, while security requirements are set on the hardware and middleware/OS of the system. This approach is consistent with the common definitions of safety and security and indicates the dependence of application process safety on system security; an insecure system cannot be safe. Furthermore, this approach enables a systematic design of ICSs that meet safety requirements for their applications exploiting appropriate security mechanisms of the underlying computing system, software or hardware.

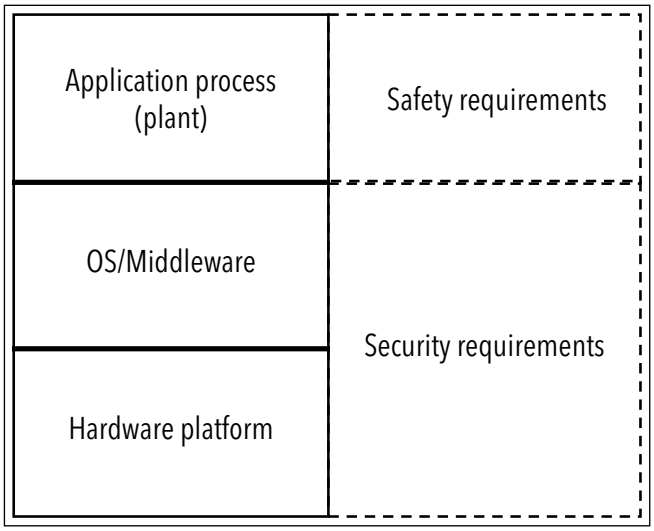

Figure 3. ICS layers.

Although we are far from an integrated approach to safety and security of ICSs and processes, there are several efforts to provide frameworks, strategies, and practices that enable their design. The International Electrotechnical Commission (IEC) 61508 standard [32] is a significant effort that addresses safety management in systems that include complex hardware and software components. Considering the limitations to prove correctness of hardware and software operations, the standard introduces a system development approach based on risk, addressing the complete lifecycle of systems, and providing technical guidelines. The standard considers the model of a system composed of a device, called equipment under control (EUC), and a control system that controls the EUC. The EUC and the control system pose risks and the standard applies to the safety requirements and management of the computing component. The basic concept of the standard is that, when designing the system, one need to identify the risks that are posed by both the control system and the EUC, determine the tolerance of each risk, and introduce safety functions that reduce all risks to tolerable levels. Importantly, the standard defines a four-grade scale of safety integrity levels (SILs) for systems, where a SIL specifies the range of probability values for a failure to perform the required safety functions; SILs are defined for two modes of system operation, continuous and on demand. Clearly, the standard specifies a framework for developing safe systems, but it does not provide or propose any specific methodologies. Its approach to system design and requirement specification is analogous to the approach for reliable computing system design. However, in reliable system design, the quantitative data related to failures are well understood and calculated, while in safe systems there are no analogous methods for quantitative arguments. It is well understood that a reliable system is not necessarily a safe system, and the difference originates from the semantic difference between a safety risk and a failure probability.

Risk-based approaches are provided by alternative efforts as well, such as the NIST guide to ICSs security [71] and the DHS recommended practice guide on improving ICS cybersecurity with defense-in-depth strategies [20]. Importantly, these efforts provide concrete techniques for the design of secure ICSs. Both the NIST and DHS guides follow an analogous approach to the IEC 61,508 approach in terms of identifying risks and taking measures to reduce them, but their focus is on the computational systems themselves than the application process that is implemented. The guides describe threats and present methods and techniques for secure computational systems, even at the level of physical security, as well as for secure networks, presenting model architectures and designs.

All risk-based approaches to the design of safe and secure ICSs identify the criticality of effective risk identification and assessment, which constitute the first steps of the design process. However, there is lack of methodologies and tools that enables the integrated analysis of risks for safety and security. Clearly, this lack originates from the dependence of safety risks on the specific (physical) process, which is controlled, and their correspondence to security risks for the computing infrastructure, which are typically described in terms of confidentiality, integrity, authentication, non-repudiation, etc. Furthermore, there is an inherent difficulty in quantifying risk for failures and attacks with the same methods, because one cannot quantify the probability of exploitation and attack when a vulnerability is discovered. It is clear, however, that there is a significant need to map or associate application (process) safety requirements with (computing) security requirements.

The need for methods to associate and map safety requirements to security ones becomes apparent when considering that violations of security at one system layer may result to violations of safety at the application layer. For example, in a PLC that controls a fluid tank and protects it from overflows, a malicious hardware or middleware attack that increments the value of the register that stores the maximum allowed fluid height may lead to an overflow and, thus, to a safety violation. Safety violations do not necessarily originate from security violations; an incorrect software implementation for example, i.e., a bug, may lead to an unsafe system state. However, security violations, in general, can lead to safety violations; and thus, security threats constitute safety threats as well.

Security threats and attacks against ICS include the ones against traditional IT systems, but there exist additional ones, because of the application domain requirements. The typical operation of ICS requires systems to meet real-time requirements and provide continuous service; violations of real-time constraints lead to application misbehavior and

unsafe states, for example, even when computations are correct. Security attacks on ICSs can be classified in two main categories, as shown in Figure 2. Computational attacks are analogous to typical IT system computational attacks, targeting computational devices, i.e., SCADA and PLCs. They include viruses, worms, Trojans, brute force access attacks, and similar, injecting code, stealing credentials, leaking data and/or hogging resources for operation disruption or DoS. Successful computational attacks result to wrong computations and application misbehavior, data leakage, violation of real-time constraints, etc. Data attacks the target communicated data. Traditional network attacks, such as man-in-the-middle and distributed denial-of-service attacks, can be launched against ICSs as well; successful attacks have similar results as computational ones. Importantly though, data attacks to ICSs include a new class of attacks, named false data injection (FDI) ones, which have not been considered in traditional IT system security. FDI attacks provide wrong data to the sensors of ICSs, leading them to wrong decisions since computations are performed on the wrong information. FDI attacks do not target the computing or network components of ICS but, rather, input false information to the overall system through its interface with the controlled physical device(s). As an example, consider an industrial system that controls a release valve of a gas tank and opens it when the measured pressure of the tank reaches a threshold; an injected (false) low value of the measured pressure in the tank may keep the release valve closed, although it should be opened, and lead to an explosion of the tank. Such attacks are feasible not only on simple systems like the gas tank, but on significantly more complex systems and processes, such as the calculation of a smart grid state, where attacks may lead to catastrophic results and operation disruption of large populations [80].

Vulnerability identification as well as risk analysis and assessment are fundamental steps in the design process of safe and secure ICS, and they have attracted significant attention in both academia and industry for appropriate tools for both the IT and OT systems. Existing tools analyze systems and networks against known risks based on: 1) the description of ICS systems at different levels of abstraction and 2) their corresponding testing and simulation results for certain risks. The CSET tool by DHS [17], for example, uses a high-level description of the system and provides lists of prioritized recommendations for improving security in the IT and OT systems based on a database of standards and guidelines provided by organizations such as INST, TSA, and DoD. It is appropriate for organization-wide assessments effectively evaluating standards and guideline compliance. There exist several security risk assessment methods that have evolved lately, especially for ICS. Many of these methods for SCADA are reviewed in [13] and include qualitative assessments, attack model-based methods as well as mathematical models for risk probability calculations, using own tools or open frameworks, such as CORAS [47]. More recently, methods have emerged that extract ICS design models through collection of the operational data and then analyze the model against threats or vulnerabilities [18]. However, the aforementioned tools do not detect some ICS-specific vulnerabilities, such as FDI attacks, and do not provide design-specific solutions against specific risks or against vulnerabilities; this is mainly because the methods are based on abstract or inadequate ICS models, which do not capture the implementation details that are the source of a specific risk or vulnerability. Clearly, there is a need for risk and vulnerability assessment methods that are automated and based on the exact implementation of a design.

Most of the risk analyses consider system and network vulnerabilities that are known from experience and the literature, but little effort has been put on the vulnerability analysis for FDI attacks. Clearly, new tools are required which will enable automated analysis for FDI attacks as well. A promising approach in this direction is the development of appropriate mathematical models for the physical processes that are implemented on an ICS and their analysis for the existence of FDI attacks. Such analyses have become feasible with recent tools, such as dReal, an SMT solver for real functions [26]. This approach has been used for the automatic vulnerability analysis of power grids to attacks that target the AC state estimation process [27]. In this case, the vulnerability analysis problem is represented as a logical decision problem, described through the state equations and the admissible sensor measurement values; the existence of input value combinations that are admissible but untrue, calculated by the solver, discloses potential successful FDI attacks. Although this approach is static, i.e., does not take into account the history of the system's operation,

it is clearly a promising method toward automated tools for vulnerability analyses of complex processes that can be described with real functions.

Designing safe and secure industrial control systems

An ICS is a hybrid system consisting of physical resources (also sometimes called plant), which are observed through sensors and are controlled by actuators through software applications. Designing such hybrid systems that are safe and secure is challenging mainly because of: 1) the hierarchy of heterogeneous subsystems, i.e., with different interfaces/platforms and by different vendors and 2) the hybrid components of ICS, i.e., the continuous physical processes and the discrete control applications. To address this challenge, the overall ICS design is logically divided into ICS computational system design and ICS network design. The former deals with the security and safety of control decisions that are implemented by computational systems and their application software, while the latter deals with the security and safety of the communicated data that deliver observations/measurements from controlled resources, as well as decisions made by the control devices. In the following sections, we discuss the two design levels, respectively.

Industrial control system security

In conventional industrial practice, the design of the computational systems and their applications, which perform calculations and control decisions, is considered secure and safe when it is shown that the system design—and the corresponding design-process workflow—complies with specific ICS standards, such as ones by the IEC and International Organization for Standardization (ISO). This is typically achieved through constructing statistical and other analysis-based risk assessment metrics and subsequently simulating and testing the system designs. However, recent attacks, such as Stuxnet [44], Aurora [77], and Mirai [55], have exposed the limitations of such practices. In fact, such standards provide only subjective measures, which are rarely objective, about the security and safety of the ICS system design. Recently, there have been efforts to develop methods that provide complimentary objective measures to design highly secure and safe ICS systems [1], [41]. These methods provide mechanized mathematical proofs that a specific ICS design is secure and safe by construction. Independently of the approach, security and safety of ICS require defenses that combine hardware and software capabilities to protect the systems against sophisticated attacks, e.g., stealthy attacks and advanced persistent threats (APTs). In the following, we present the current trends in industry and recent research results to handle security and safety of ICS systems at design level.

Current trends

The goal of current industrial practices is to show that an ICS system design meets the required security and safety standard as defined by various standardization bodies, e.g., ISO [35], ICS-CERT [30], and IEC [31]. These standards provide the general guidelines to identify the risk of various hazards, attacks, and threats [36] and also provide guidelines for the protection of the systems against such attacks/threats or to reduce the impact of the risk. To show that a given ICS design adheres to the required guidelines, system designers usually construct several metrics based on the testing and simulation of a given design [15]. More recently, a statistically rigorous method for testing ICS systems has been proposed [62]. The method employs high-throughput testing combinatorial methods, which enables multifacet testing and analysis of cyber threats by creating a probabilistic model of the system's response. Furthermore, the method also helps to determine optimum defense configurations for system resilience. However, such methods fail to assure that a given design is free of some classes of attacks, vulnerabilities, and threats, such as data integrity attacks, and cannot identify sophisticated threats to ICS systems, e.g., APTs, as demonstrated by recent attacks.

Since the development of an ICS design, which employs a distributed supply chain process that involves various organizational divisions and vendors, there is a high risk that the process may introduce some vulnerabilities in the designed system. There are two aspects in the security of this process. The first one is to secure the supply chain itself, protecting it from misuse. The second one is to develop techniques that enable the integration of components in a secure system, although the components originate from different groups (with different tools, design techniques, and testing processes) and may have vulnerabilities; i.e., we need techniques that integrate untrusted components. To address the

first issue, industries employ standards that provide guidelines for various phases of the design [34]. Recently, there have been several efforts to improve the reliability of the ICS design process based on cryptographic techniques; for example, Healthchain [4] is developed based on a blockchain technology and enables cost-effective and trusted trade among various vendors. For the second issue, several techniques have been developed for hardware and software components. A promising technique exploits an integrated trusted protection model [75], which provides a trusted computing platform that protects data sharing among components through trusted data protection and network management mechanisms. Such a trusted platform enables protection of components at different layers of an ICS network taking into account the specific security and differentiated services requirements of each layer. Software components of ICS employ different techniques, to establish system integrity and process isolation and to protect the design process; such techniques include secure system booting [5] and process level attestation techniques [51]. These methods also help to manage distributed control of the processes, e.g., memory management and interprocess communication [46], [81]. Overall, the increasing automation of these processes requires more robust software security techniques. Software techniques have several advantages over hardware techniques, e.g., cost, continuous evolution, and dynamic changes. Furthermore, the combination of software techniques with trusted computing modules enables the development of trusted computing platforms for applications and services [59]. Although these approaches achieve high reliability of ICS system design, they fail to assure that the designed ICS system adheres to all required standards and that it is free of built-in vulnerabilities. Therefore, a lot of effort is being spent recently to develop methods that assure that an ICS design is secure and safe by construction; these methods employ formal verification techniques as we describe in the following.

Secure and safe industrial control system design

Recent developments in the area of formal verification have enabled the development of secure and safe ICS design by construction [21], [38]. Such approaches have been widely used in other domains, such as implementation of web applications and cryptographic algorithms [14], [23], [76], and recently they have been adopted for designing ICS [41], [63]. Techniques have been developed to model cyber (discrete) and physical (continuous) resources of ICS, which are heterogeneous in characteristics and semantics [41]. Based on the adequate ICS models/specification, ARMET [19], [41] derives secure and reliable implementation through stepwise but sound refinements using a theorem prover, Coq, by employing deductive program synthesis techniques. Each refinement assures (based on a mathematical proof) that no additional behavior or vulnerability has been introduced in the designed ICS. Such an approach assures that ICS design is secure and safe by construction.

In another effort [3], the ROSCoq framework has been developed using the theorem prover Coq to model cyber and physical resources of robots. The models are formalized in an extended logic of events combined with CoRN theory of constructive real analysis. Based on the model, the framework enables the proof of security and safety properties of the model. Furthermore, a Coq library "VeriDrone" [12], [49] has been developed to reason about security of ICS models at different and independent levels. The library enables the derivation of an ICS implementation in C from high-level ICS models. Such approaches derive highly assured implementations from the developed ICS design, but their application heavily relies on adequate modeling of cyber and physical resources of ICS design, which usually requires significant effort and time.

Industrial network security

In typical ICSs, systems are interconnected over industrial networks, which implement specialized protocols, e.g., EtherNet/IP, Modbus/TCP, Sinec H1, Profibus, CANopen, and Fieldbus [33]. These protocols meet different functional requirements, have different interfaces, and lead to different system requirements in terms of memory, energy, and processing [25]. In contrast to other networks, the primary function of industrial networks is to control physical processes and resources, implementing a hierarchy of networks, as shown in Figure 2, which implement different protocols and standards at the various levels of the hierarchy. A conventional ICS network has a supervisory network at the top of the hierarchy that may be also connected to the Internet. These networks are further connected to controller networks, which have connections to field

equipment subnetworks [25]. Failures of industrial networks have higher cost, because of the resulting process disruptions, and thus, industrial networks require more reliability relatively to other networks. Furthermore, industrial networks differ from other typical IT networks in that they have small packets that support synchronous and asynchronous communication among interconnected devices, which often reside in environments with hostile conditions, such as high dust, heat, and vibration. Importantly, industrial networks adhere to strict real-time communication requirements, requiring low overhead at each component. For this purpose, industrial networks typically employ short protocol stacks that involve a few layers, e.g., three layers, as shown in Figure 4 from [25], with the application, data link, and physical layers. This approach leads to low communication delays and enables the adoption of variants of standard network protocols, e.g., customised TCP, to achieve real-time communication among ICS components. Often, ICS networks include the functionality of network layer in the application layer. These characteristics and constraints of ICS network devices lead to several security challenges for the networks, because they make them more vulnerable to malicious activities than typical IT networks.

During the early period of industrial automation, ICS networks were protected following the "security through obscurity" approach, because physical resources were separate from cyber resources. The main threats at that time were system integrity attacks, e.g., undesired interference by an unhappy worker [22]. With the recent developments in the automation of industrial control process, new vulnerabilities and threats to ICS networks have emerged because control systems have become accessible through networks. Such increased connectivity immediately brought attacks through the spread of malicious software as well as failures because of unreliable networks due to third party services [67].

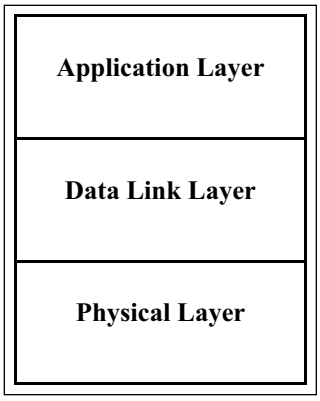

Figure 4. IEC 61158 network stack for industrial networks.

Recently, new threats have emerged to ICS networks, such as APTs, where adversaries attack selected target systems, as evidenced by Stuxnet [44]. Clearly, the sophistication of Stuxnet attack is an evidence that the attackers had detailed knowledge of the control system. Due to such attacks, industry has started adapting more sophisticated security strategies such as the "defense in depth" approach proposed by NIST [70], where security mechanisms are implemented at every layer of control networks, protecting them against external threats.

One of the desired goals of ICS networks is to ensure that the data are communicated only among authorized components (e.g., users, processes, or nodes) and that the exchanged data are "legal," i.e., correct in the given component context. In general-purpose networks, this is achieved through different techniques, e.g., by implementing access control using firewalls at application and network layers or in the network infrastructure [10]. Firewalls can be easily configured and adapted for ICS networks because these networks have well-defined communication interfaces. However, the implementation level of firewalls depends on the hierarchical level of an ICS (sub) network; an *ad hoc* sensor network, for example, may need protection at the data link level, while the centralized supervisory network may need protection at network level [68].

Access control is not sufficient to ensure that ICS network services are continuously available. Hence, it is important to protect ICS network services against malicious activity to ensure their continuous availability without hindering their performance [57]. Such protection of the network can be achieved by hardening the security of hardware components and keeping the software components up-to-date. However, such hardening cannot prevent insider attacks, where an authorized person acts maliciously. Such threats can be handled at the organizational level by recording component actions and auditing them to detect such threat.

Secure communication in general purpose networks is typically achieved by cryptographic primitives for encryption and authentication, such as RSA

[65] and AES [2]. However, these solutions cannot be directly employed in ICS environments, because they require significant resources, processing, and memory. Elliptic cryptography-based solutions have been developed recently to enable secure communication in environments with low computational resources and provide high security, comparable to alternative public key cryptographic methods [52]. Also, considering the specific requirements of industrial networks, NIST has developed SHA-3, a set of hash functions for the generation of keys, pseudorandom bits, and digital signatures for resource-variant networks [53].

ICSs rely heavily on sensor networks to control and manage physical resources/processes. Sensor networks are characterized by their limited node computational resources and the ad hoc connectivity of a large number of nodes. As sensor networks have very strict performance requirements, their security implementation, typically through cryptographic mechanisms at the link layer, is also demanding in terms of performance and resources. To ensure secure communication in these resource-limited networks, there are approaches where the employed encryption mechanisms have different levels of complexity to communicate different data, choosing the appropriate level of encryption based on the value of the communicated information [78].

In cryptographic mechanisms, the key management plays a vital role to ensure security of the communication among ICS components. Weaknesses in key management may lead to compromised communication independently of the strength of the underlying encryption scheme. Global keys cannot be used in networked systems because an isolated compromise of network security may lead to network-wide compromised keys. Therefore, secure ICS networks require effective mechanisms to generate and distribute keys. One such mechanism allows to use temporary global keys and a global permanent key to construct a main key; after distributing the main key, it destroys the temporary global key to avoid key leakage [61]. Alternatively, one can employ random key distribution, which requires a sufficient set of keys, so that all end points of a network can communicate securely [11].

In ICSs, the supervisory network is mainly responsible for controlling the overall industrial process. Lately, emerging process control models connect the supervisory network to the Internet as well; this makes the entire ICS network highly vulnerable to network attacks, including DoS and distributed DoS (DDoS) attacks. These attacks disrupt network systems/services by overloading system resources (e.g., memory, processing, and energy), so that they fail to perform their intended functionality and to respond to other nodes/users in a timely fashion. There are two main variations of these attacks. The first type aims to exploit hardware or software vulnerabilities by sending packets to crash the target system. This type of attack is carefully organized and constructs packets appropriately, e.g., to exploit an identified vulnerability, before sending them to the network [29]. As ICS network components are often not patched, such vulnerabilities can be easily exploited; this originates from the fact that ICS network components are typically not configured for automatic software update, in order to avoid operational disruptions during the update process. In the second type of the attacks, DDoS attacks, a large number of systems create excessive amounts of network traffic—in addition to legitimate traffic—targeting the victim system. This traffic heavily overloads the resources of the target system, making it significantly less responsive and leading it to fail to serve its legitimate users. The recent Mirai incident [55] demonstrates that industrial systems are vulnerable to such attacks that are launched through malware injection.

Defending against DDoS attacks is a very difficult task, because these attacks exploit shared network components that are accessible by all connected systems. Usually, intrusion detection and filtering mechanisms are used to trace such attacks [60]. Intrusion detection mechanisms employ signature and anomaly based detection techniques [74], while filtering ones employ packet marking and logging techniques for detection and attack tracing, respectively [66], [69]. Considering the heterogeneous characteristics of continuous operation of ICS subnetworks that keep evolving, it is important to manage the security of such networks automatically and autonomously, in contrast to conventional IT network management approach that is based on configuration setting. To address the security and safety, such networks are continuously monitored through different mechanisms, which raise alarms when some unexpected occurs, as described in the following section.

Monitoring industrial control systems for security and safety

Independently of compliance with standards, an ICS that has been designed to be safe and secure is not guaranteed to operate as expected in a real environment; ICS, like other systems, are designed and tested in a safe and controlled environment, while their field operation is in larger and uncontrolled environments. Thus, it is necessary to monitor system operation at runtime to ensure that the process execution is secure and safe as well. This monitoring is typically implemented with security monitors, which compare system execution with a reference model to detect any security or safety breach. There are two facets of monitoring in ICS environments: 1) monitor the identity and privacy of the ICS system components and users, assuring that only legal components/users have access to ICS system and 2) monitor the reliability of the operations of the ICS, assuring that ICS components/users are operating as desired. Based on these facets, in the following, we present methods to monitor ICSs.

Unauthorized access

One of the critical threats to ICS is compromised by illegitimate system components or users. It is necessary to ensure that all components and users who exchange information in the system are legitimate ones, taking into account even legal requirements in some application domains, such as smart grid and medical systems. Access control mechanisms can be implemented in different ways to allow only legal components/user to access system components. The decision for a specific implementation depends on the profile of the component that needs to be protected against unauthorized access. For example, the identity of human users can be established through an access control mechanism that employs text-based methods, e.g., pin codes and account/password login credentials, or biometric methods, e.g., iris scans, fingerprints, and face patterns. The privacy of a user can be established by ensuring that the user is the only one that has been granted privilege to access a specific component. Furthermore, the identity of devices and applications can be established in many ways, e.g., based on their unique MAC address, IP address, port numbers, etc. The privacy of devices and applications can be established by control mechanisms that employ, for example, digital certificates and encryption techniques. Considering the diverse characteristics of ICS components and users, a popular strategy is to employ control mechanisms that describe access policies at different levels of abstraction. Recently, a role-based access control mechanism has been proposed, which allows modeling high-level policy descriptions and low-level access details, including system structure and architectural details [37]. The approach ensures consistency among policies by specifying them as logical formulas and verifying their correctness. Besides describing correct and consistent policies, it is also important to enforce such policies at runtime. There have been several efforts to automatically enforce access control policies at runtime in general purpose computing domains, e.g., the Jeeves language enables run-time enforcement of information flow policies by construction [76]. Recently, a more promising solution has been developed to enforce access control and other security policies through the ARMET run-time monitor [41]. The ARMET approach allows designers to express requirements for legitimate use, e.g., access control and policies, as conditions in an ICS at any desired level of abstraction; i.e., they can be expressed as preconditions, postconditions, or invariants in a program. For example, a module that is used by unclassified users can be restricted from accessing specific variables that are available only to classified ones. Such requirements can be specified in ARMET and refined into conditions that are enforced by a run-time monitor, which detect all violations of the defined conditions. Importantly, the ability to program policies as conditions enables ICS evolution by dynamically adapting the security monitors to check new conditions, as they emerge in standards and directives.

Enforcing access control policies through appropriate mechanisms ensures that only legitimate users and components have the intended access permissions to all resources of the ICS. However, such access controls and policies cannot prevent components from performing malicious activity by exchanging undesired information over the network, which may compromise other network resources by alternate means, e.g., insider attacks, such as the case of Snowden [73], and APTs, such as Stuxnet. So, in order to protect ICS networks and components against malicious activity, security monitors have been developed which monitor the systems at runtime to detect any such malicious activity.

Intrusion detection/prevention

Resource-variant ICS environment has well-defined communication; and thus, sophisticated attacks can be detected through monitoring the network traffic among system components. Intrusion detection systems (IDSs) are an example of such monitors, which aim to detect attacks by understanding system behavior through analysis of network traffic. There are different variants of IDS which depend on: 1) how such systems characterize the behavior (e.g., profile-based or model-based) and 2) how they compare behaviors (e.g., comparison to bad behavior or violation of good behavior). These variants of monitors can be classified into four classes, as shown in Figure 5.

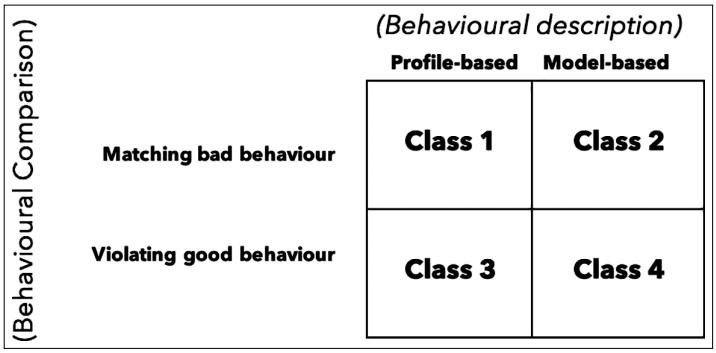

Figure 5. IDS classification.

Profile-based methods build a profile of system components by observing system parameters. Based on the profile, Class 1 monitors attempt to detect malicious activities by matching the system behavior with known bad behavior. These monitors employ statistical methods to build profile of bad behaviors and attacks [28], [72]. These monitors are robust relatively to model-based monitors (Class 2), because machine learning methods are capable to detect more generic attacks, but they are practically limited by high rates of false alarms and do not provide adequate information for diagnosis. Class 3 monitors build profile(s) of good behavior of system components and detect violation to them [42], [43]. These monitors (Class 3) are more robust than Class 1 monitors because they do not require any previous information of attacks and, thus, can detect new attacks. Although these monitors are capable to detect any deviation from good behavior and report the violation as an attack, they raise false alarms because deviation can just be accidental or could be part of a normal but rare behavior. Additionally, these monitors provide very limited information for diagnosis because they only know that something different from "good behavior" has happened.

Classes 2 and 4 systems offer model-based monitoring. They are very popular in highly secure and critical environments where security failures have a high cost. Based on the reference behavioral model of the observed system, these monitors provide very rich information for diagnosis when malicious activity has been detected. However, Class 2 monitors are limited in that they can detect only known attacks by comparing models of bad behaviors, e.g., signature-based IDS [58], [64]. Class 4 monitors can provide even higher diagnostic information because they know exactly what part of the model has been violated. Since the monitors passively evaluate network traffic and raise alarms when a threat is detected, they suffer from false negatives, failing to block attacks actively. Intrusion protection systems (IPSs) have been developed to block attacks when they detect them [24]. However, IPS suffers from high false positives, since they may block legal behavior that is not predicted by the models, leading to unnecessary ICS operation disruptions. Despite the strengths of these passive and active monitors, there is an execution overhead to compare high granularity models, which may hinder runtime performance of ICS networks. Furthermore, the aforementioned monitors suffer from false alarms, which make them ineffective for ICS environments. Recently, there has been effort to develop monitors based on behavioral models, which are efficient and rigorous, i.e., they meet real-time requirements and are free of false alarms, as we present below.

Behavior-based monitoring

Behavior-based monitoring applies logical specification to describe the behavior of the target system. Based on such behavioral specification, ARMET [41] provides a unified method to design, implement, and monitor ICS systems. ARMET is a framework that has three basic components, which allow: 1) to build ICS that are secure and safe by design; 2) to monitor the operations of ICS in real-time for safety and security; and 3) to recover system execution into a safe state, in the case of attack or failure. ARMET has been developed for the high assurance security and safety of ICS and cyber-physical systems, in general. The approach can be applied to other embedded systems as well, due to their complexity equivalence.

To develop an ICS application, ARMET initially requires an executable specification of the application, which is proven to be consistent with defined security and safety application properties. The process to develop such a specification is based on Fiat [19], which applies deductive synthesis to develop such specification through stepwise but sound refinements; the proofs are carried in the theorem prover Coq. Furthermore, the specification includes the first class models of cyber and physical resources of an ICS coupled with their functional and non-functional properties. Furthermore, the specification includes description of normal behavior, also denoted "good behavior" as well as compromised application behavior, also denoted as "bad behavior" including known attacks. Through program synthesis, the application implementation can be generated automatically from the given specification, which is provably safe and secure, meeting the requirements of the specification. At the end of this process, ARMET has specification and implementation of the required ICS system.

ARMET monitors the behavior of the application, comparing the application execution behavior to the specified good application behavior. Specifically, ARMET runs the executable specification and the implementation of the application in parallel and observes their consistency. As shown in Figure 6, the ARMET middleware has various components whose core is run-time security monitor (RSM). The other auxiliary components include diagnosis and recovery modules, trust model, backup, and selection module. The RSM takes the application specification and implementation as inputs and executes them in parallel. The monitor generates "predictions" by specification execution and produces "observations" by executing the implementation. Since normal and compromised behaviors are described in the specification, the monitor can detect all violations of normal ("good") behavior and compromised ("bad") behavior, whether originating from attack or failure. The RSM has been proven to be sound and complete [39], [40], asserting that the monitor is free of false alarms, positive, or negative. This is a highly desired property in practical ICS, where false alarms may lead to significant loss of resources and productivity.

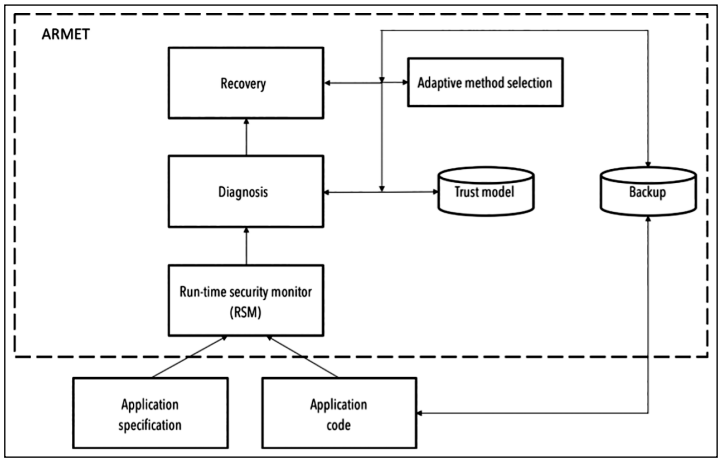

Figure 6. The ARMET architecture.

For recovery, when an alarm is raised by the RSM, the diagnosis module of ARMET attempts to identify the failure or attack based on the context as determined by the trust module. Based on the diagnosed cause, the recovery module attempts to recover the application into a safe state by finding alternative resources (e.g., computational libraries and data) to the compromised ones. In the case of unavailability of alternate resources, the application is recovered into its last safe state from backup.

Experiments have shown that the RSM efficiently detects violations and failures by meeting real-time performance requirements of ICS applications [41]. Furthermore, the ARMET approach requires that the application specification is executed in a safe environment and cannot be compromised, so that predictions are correct. This assumption can be realized with existing methods, such as trusted platforms like Intel SGX [16] and ARM TrustZone [6].

Another similar approach has developed a method to detect attacks on ICS for water treatment plants by inferring invariants of physical processes through observation of the ICS at runtime and checking their consistency with the invariant that were set during various stages of ICS development [1]. An alternative effort has developed a reasoning framework to detect attacks in ICS environments [63]. This framework allows to model the physical process of an ICS by considering their physical interactions and time and state discretization. The models are state transitions systems developed in the language ASLan++ and are analyzed by the corresponding tool CL-AtSe. The framework has been successful in detecting real-time attacks in a testbed of ICS for water treatment plant. However, these approaches are limited for practical use due to the verification techniques which are not scalable and suffer from state explosion.

The design of safe and secure ICSs is a challenging task. The strong requirements on ICSs for safety, continuous operation, and real-time constraints make their design more demanding than that of the traditional IT systems. Importantly, ICSs are vulnerable not only to the traditional threats of IT systems but to a larger set of threats, such as FDI attacks. Successful attacks on ICSs do not necessarily need to violate correctness of computations; they can lead systems to wrong decisions through FDI attacks, to safety violation by overloading the system and leading it to violate real-time constraints, etc. Despite the significant progress of the past few decades in the development of methods and techniques for safe and secure ICSs, it is clear that we are in the beginning of an effort that faces many design challenges. The technical area of safe and secure ICSs provides a challenge and an opportunity for new approaches, methods, and techniques for several years to come.

References

[1] S. Adepu and A. Mathur, "Using process invariants to detect cyber attacks on a water treatment system," in *ICT Systems Security and Privacy Protection. SEC 2016. IFIP Advances in Information and Communication Technology*, J. H. Hoepman and S. Katzenbeisser, Eds. Springer, 2016, vol. 471.

[2] NIST, "Advanced encryption standard," FIPS Publication 197, Nov. 26, 2001.

[3] A. Anand and R. Knepper, *ROSCoq: Robots Powered by Constructive Reals*. Springer, 2015, pp. 34–50.

[4] T. Ahram, A. Sargolzaei, S. Sargolzaei, J. Daniels, and B. Amaba, "Blockchain technology innovations," in *Proc. IEEE Technol. Eng. Manage. Conf. (TEMSCON)*, Santa Clara, CA, USA, 2017, pp. 137–141.

[5] W. Arbaugh, D. Farber, and J. Smith, "A secure and reliable bootstrap architecture," in *Proc. IEEE Symp. Security Privacy*, 1997, pp. 65–71.

[6] ARM Security Technology, "Building a secure system using TrustZone technology," ARM white paper, Document PRD29-GENC-009492C, 2005. [Online]. Available: http://infocenter.arm.com/help/topic/com.arm.doc.prd29-genc-009492c/PRD29-GENC-009492C_trustzone_security_whitepaper.pdf

[7] Y. Bertot and P. Castran, *Interactive Theorem Proving and Program Development-Coq'Art: The Calculus of Inductive Constructions. Springer,* 2004.

[8] T. S. Bernard, T. Hsu, N. Perlroth, and R. Lieber, "Equifax says cyberattack may have affected 143 million in the U.S.," *The New York Times*, Sept. 7, 2017.

[9] D. Bisson, "10 of the most significant ransomware attacks of 2017," *Tripwire*, Dec. 10, 2017. [Online]. Available: https://www.tripwire.com/state-of-security/security-data-protection/cyber-security/10-significant-ransomware-attacks-2017/

[10] D. Bolding, "Network security, filters and firewalls," *Crossroads*, vol. 2, no. 1, pp. 8–10, 1995.

[11] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *Proc. IEEE Symp. Security Privacy*, 2003, pp. 197–213.

[12] M. Chan, D. Ricketts, S. Lerner, and G. Malecha, "Formal verification of stability properties of cyber-physical systems," in *Proc. CoqPL'16*, Jan. 2016.

[13] Y. Cherdantseva et al., "A review of cyber security risk assessment methods for SCADA systems," *Comput. Secur.*, vol. 56, pp. 1–27, 2016.

[14] A. Chlipala, "Ur/web: A simple model for programming the web," *Commun. ACM*, vol. 59, no. 8, Aug. 2016.

[15] Z. A. Collier, M. Panwar, A. A. Ganin, A. Kott, and I. Linkov, "Security metrics in industrial control systems," in *Cyber Security of Industrial Control Systems, Including SCADA Systems*, New York, NY, USA: Springer, 2016.

[16] V. Costan and S. Devadas, "Intel SGX explained," Cryptology ePrint Archive: Report 2016/086, IACR, 2016.

[17] DHS/ICS-CERT, "Cyber security evaluation tool." [Online]. Available: https://cset.inl.gov/SitePages/Home.aspx

[18] CyberX, "Automated ICS threat modeling." [Online]. Available: https://cyberx-labs.com/en/proactively-mitigate-ics-risk-ics-attack-vector-predictions/

[19] B. Delaware, C. Pit-Claudel, J. Gross, and A. Chlipala, "Fiat: Deductive synthesis of abstract data types in a proof assistant," in *Proc. 42nd Annu. ACM SIGPLAN-SIGACT Symp. Principles Programming Languages (POPL'15)*, Mumbai, India, Jan. 15–17, 2015, pp. 689–700.

[20] DHS Recommendations, "Recommended practice: Improving industrial control system cybersecurity with defense-in-depth strategies," Industrial Control Systems Cyber Emergency Response Team, 2016. [Online]. Available: https://ics-cert.us-cert.gov/sites/default/files/recommended_practices/NCCIC_ICS-CERT_Defense_in_Depth_2016_S508C.pdf

[21] "The Science of Deep Specification," NSF Project, 2016–2021 [Online]. Available: https://deepspec.org/main

[22] D. Dzung, M. Naedele, T. P. Von Hoff, and M. Creavtin, "Security for industrial communication systems," *Proc. IEEE*, vol. 93, no. 6, pp. 1152–1177, Jun. 2005.

[23] A. Erbsen, "Crafting certified elliptic curve cryptography implementations in Coq," M. Eng. thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 2017. [Online]. Available: https://dspace.mit.edu/handle/1721.1/112843

[24] A. Fuchsberger. *Intrusion Detection Systems and Intrusion Prevention Systems*. Inf. Secur. Tech. Rep. 10, Jan. 3, 2005, pp. 134–139.

[25] B. Galloway and G. P. Hancke, "Introduction to industrial control networks," *IEEE Commun. Surveys Tutorials*, vol. 15, no. 2, pp. 860–880, Second Quarter 2013.

[26] S. Gao, S. Kong, and E. M. Clarke, "dReal: An SMT solver for nonlinear theories over the reals," in *Proc. Int. Conf. Autom. Deduction*, 2013, pp. 208–214.

[27] S. Gao, L. Xie, A. Solar-Lezama, D. Serpanos, and H. Shrobe, "Automated vulnerability analysis of ac state estimation under constrained false data injection in electric power systems," in *Proc. 54th IEEE Conf. Decision Control (CDC)*, Osaka, Japan, 2015, pp. 2613–2620.

[28] V. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artif. Int. Rev.*, vol. 22, no. 2, pp. 85–126, Oct. 2004.

[29] A. Hussain, J. Heidemann, and C. Papadopoulos, "A framework for classifying denial of service attacks," in *Proc. Conf. Appl. Technol. Archit. Protocols Comput. Commun. (SIGCOMM '03)*, New York, NY, USA, 2003, pp. 99–110. [Online]. Available: http://dx.doi.org/10.1145/863955.863968

[30] "Industrial Control Systems Cyber Emergency Response Team (ICS-CERT)." [Online]. Available: https://ics-cert.us-cert.gov/Standards-and-References

[31] IEC 61158, "Industrial automation systems and integration," 2003. [Online]. Available: https://www.iso.org/obp/ui/#iso:std:iso:15745:-3:ed-1:v1:en

[32] "Functional safety and IEC 61508 Standard." [Online]. Available: http://www.iec.ch/functionalsafety/

[33] "IPFS list of industrial communication protocols." [Online]. Available: https://ipfs.io/ipfs/QmXoypizjW3WknFiJnKLwHCnL72vedxjQkDDP1mXWo6uco/wiki/List_of_automation_protocols.html

[34] ISO Supply Chain Security Standards, "ISO supply chain management standards to reduce risks of terrorism, piracy and fraud," 2007. [Online]. Available: https://www.iso.org/news/2007/10/Ref1086.html

[35] ISO/IEC TR 27019:2013, "Information technology—Security techniques—Information security management guidelines based on ISO/IEC 27002 for process control systems specific to the energy utility industry," 2013. [Online]. Available: https://www.iso.org/standard/43759.html

[36] ISO Standard, "Robots and robotic devices—Safety requirements for personal care robots," 2014. [Online]. Available: https://www.iso.org/standard/53820.html

[37] I. C. Bertolotti, L. Durante, L. Seno, and A. Valenzano, "A twofold model for the analysis of access control policies in industrial networked systems," *Comput. Stand. Inter.*, vol. 42, no. C, pp. 171–181, Nov. 2015.

[38] D. Kästner, J. Barrho, U. Wünsche, M. Schlickling, B. Schommer, M. Schmidt, et al. "CompCert: Practical experience on integrating and qualifying a formally verified optimizing compiler," in *Proc. ERTS2—Embed. Real Time Soft. Syst.*, Toulouse, France, Jan. 2018.

[39] M. T. Khan, D. Serpanos, and H. Shrobe, "On the formal semantics of the cognitive middleware AWDRAT," Tech. Rep. MIT-CSAIL-TR-2015-007, Computer Science and Artificial Intelligence Laboratory, MIT, USA, Mar. 2015.

[40] M. T. Khan, D. Serpanos, and H. Shrobe, "A rigorous and efficient run-time security monitor for real-time critical embedded system applications," in *Proc. IEEE 3rd World Forum Internet Things (WF-IoT)*, Reston, VA, 2016, pp. 100–105.

[41] M. T. Khan, D. Serpanos, and H. Shrobe, "ARMET: Behavior-based secure and resilient industrial control systems," *Proc. IEEE*, vol. 106, no. 1, pp. 129–143, Jan. 2018. doi: 10.1109/JPROC.2017.2725642

[42] S. S. Kim, A. L. N. Reddy, and M. Vannucci, "Detecting traffic anomalies through aggregate analysis of packet header data," in *Proc. 3rd Int. IFIP-TC6 Network. Conf. (NETWORKING 2004)*, Athens, Greece, Springer LNCS-3042, May 9–14, 2004, pp. 1047–1059.

[43] A. Lakhina, M. Crovella, and C. Diot, "Mining anomalies using traffic feature distributions," in *Proc. Conf. Appl. Technol. Archit. Protocols Comput. Commun. (SIGCOMM 2005)*, Philadelphia, PA, USA, Aug. 22–16, 2005, pp. 217–228.

[44] R. Langner, "Stuxnet: Dissecting a cyberwarfare weapon," *IEEE Secur. Priv.*, vol. 9, no. 3, pp. 49–51, May 2011.

[45] N. G. Leveson, *Safeware—System Safety and Computers*. Addison-Wesley Professional, 1995.

[46] D. Lie, C. Thekkath, and M. Horowitz, "Implementing an untrusted operating system on trusted hardware," *ACM SIGOPS Operating Syst. Rev.*, vol. 37, no. 5, pp. 178–192, 2003.

[47] M. S. Lund, B. Solhaug, and K. Stolen, *Model-Driven Risk Analysis—The CORAS Approach*. Berlin, Heidelberg: Springer-Verlag, 2011.

[48] M. W. Maier, "Architecting principles for systems-of-systems," in *Proc. 6th Annu. INCOSE Symp.*, Boston, MA, USA, 1996, pp. 567–574.

[49] G. Malecha, D. Ricketts, M. M. Alvarez, and S. Lerner, "Toward foundational verification of cyber-physical systems," in *Proc. Sci. Security Cyber-Physical Syst. Workshop (SOSCYPS)*, Apr. 2016, pp. 1–5.

[50] D. McMillen, "Attacks targeting industrial control systems (ICS) up 110 percent," SecurityIntelligence, Dec. 27, 2016. [Online]. Available: https://securityintelligence.com/attacks-targeting-industrial-control-systems-ics-up-110-percent/

[51] MICROSOFT, "Shared source initiative," 2011. [Online]. Available: http://www.microsoft.com/resources/ngscb/default.mspx

[52] V. S. Miller, "Use of elliptic curves in cryptography," in *Proc. Adv. Cryptology (CRYPTO '85)*, Springer-Verlag, London, U.K., 1985, pp. 417–426.

[53] M. J. Dworkin, "SHA-3 standard: Permutation-based hash and extendable-output functions," Federal Information Processing Standards (NIST FIPS)–202, Aug. 4, 2015.

[54] L. H. Neuman, "GitHub survived the biggest DDoS attack ever recorded," Wired, Mar. 1, 2018. [Online]. Available: https://www.wired.com/story/github-ddos-memcached/

[55] L. H. Newman, "What we know about Friday's massive east coast internet outage," Wired, Oct. 21, 2016.

[56] NIST, "Framework for cyber-physical systems—Release 1.0," May 2016. [Online]. Available: https://s3.amazonaws.com/nist-sgcps/cpspwg/files/pwgglobal/CPS_PWG_Framework_for_Cyber_Physical_Systems_Release_1_0Final.pdf

[57] T. Novak and A. Gerstinger, "Safety- and security-critical services in building automation and control systems," *IEEE Trans. Industrial Electron.*, vol. 57, no. 11, pp. 3614–3621, Nov. 2010.

[58] V. Paxson, "Bro: A system for detecting network intruders in real-time," *Comput. Netw.*, vol. 31, no. 23–24, pp. 2435–2463, 1999.

[59] S. Pearson, *Trusted Computing Platforms: TCPA Technology in Context*. Prentice Hall, 2002.

[60] T. Peng, C. Leckie, and K. Ramamohana-Rao, "Survey of network-based defense mechanisms countering the DoS and DDoS problems," *ACM Comput. Surv.*, vol. 39, no. 1, Article 3, 2007.

[61] A. Perrig, J. Stankovic, and D. Wagner, "Security in wireless sensor networks," *Commun. ACM*, vol. 47, no. 6, pp. 53–57, 2004.

[62] O. U. Rehman and K. F. Joiner, "Test strategy to detect industrial control systems' common cyber weaknesses and vulnerabilities," *INCOSE Int. Symp.*, vol. 27, pp. 796–810, 2017.

[63] M. Rocchetto and N. O. Tippenhauer, "Toward formal security analysis of industrial control systems," in *Proc. ACM Asia Conf. Comput. Commun. Security (ASIA CCS '17)*, New York, NY, USA, 2017, pp. 114–126.

[64] M. Roesch, "Snort—Lightweight intrusion detection for networks," in *Proc. 13th USENIX Conf. Syst. Admin. (LISA '99)*, 1999, pp. 229–238.

[65] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120–126, Feb. 1978.

[66] S. Savage, D. Wetherall, A. Karlin, and T. Anderson, "Network support for IP traceback," *IEEE/ACM Trans. Network.*, vol. 9, no. 3, pp. 226–237, 2001.

[67] C. Schwaiger and A. Treytl, "Smart card based security for fieldbus systems," in *Proc. IEEE Conf. Emerging Technol. Factory Autom.*, vol. 1, Sept. 2003, pp. 398–406.

[68] S. Slijepcevic, M. Potkonjak, V. Tsiatsis, S. Zimbeck, and M. Srivastava, "On communication security in wireless *ad hoc* sensor networks," in *Proc. 11th IEEE Int. Workshop Enabling Technol.*, 2002, pp. 139–144.

[69] A. Snoeren, et al., "Single-packet IP traceback," *IEEE/ACM Trans. Networking*, vol. 10, no. 6, pp. 721–734, 2002.

[70] K. Stoufer, J. Falco, and K. Scarfone, "Guide to industrial control systems (ICS) security," National Institute of Standards and Technology, Final Public Draft, Sept. 2008.

[71] K. Stouffer, S. Lightman, V. Pillitteri, M. Abrams, and A. Hahn, "Guide to industrial control systems (ICS) security, National Institute of Standards and Technology," SP 800-82 Rev. 2, 2015. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-82r2.pdf

[72] A. Valdes and K. Skinner, "Adaptive, model-based monitoring for cyber attack detection," in *Proc. 3rd*

Int. Workshop Recent Advan. Intrusion Detect. (RAID 2000), Toulouse, France, Oct. 2–4, 2000, pp. 80–93.

[73] J. Verble. "The NSA and Edward Snowden: Surveillance in the 21st century," *SIGCAS Comput. Soc.*, vol. 44, no. 3, pp. 14–20, Oct. 2014.

[74] H. Wang, D. Zhang, and K. Shin, "Detecting SYN flooding attacks," in *Proc. 21st Annu. Joint Conf. IEEE Comput. Commu. Soc. (INFOCOM'02)*, 2002, pp. 1530–1539.

[75] J. Wang, J. Liu, S. Yang, and M. Zhang, "Integrated trusted protection technologies for industrial control systems," in *Proc. 18th Int. Conf. Advan. Commun. Technol. (ICACT)*, Pyeongchang, 2016, pp. 70–75.

[76] J. Yang, K. Yessenov, and A. Solar-Lezama, "A language for automatically enforcing privacy policies," in *Proc. 39th ACM Symp. Principles Program. Lang. (POPL 2012)*, Philadelphia, PA, USA, Jan. 25–27, 2012, pp. 85–96.

[77] M. Zeller, "Myth or reality—Does the aurora vulnerability pose a risk to my generator?" in *Proc. 64th Annu. Conf. Protective Relay Eng.*, Apr. 2011, pp. 130–136.

[78] S. Zhu, S. Setia, and S. Jajodia, "LEAP: Efficient security mechanisms for large-scale distributed sensor networks," in *Proc. 10th ACM Conf. Comput. Commun. Sec.*, 2003, pp. 62–72.

[79] B. Kang, K. McLaughlin, and S. Sezer, "Towards a stateful analysis framework for smart grid network intrusion detection," in *Proc. 4th Int. Symp. ICS & SCADA Cyber Secur. Res.*, 2016, pp. 1–8.

[80] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," *ACM Trans. Inform. Syst. Sec.*, vol. 14, no. 1, pp. 1–33, 2011.

[81] T. Garfinkel, M. Rosenblum, and D. Boneh, "Flexible OS support and applications for trusted computing," in *Proc. 9th Conf. Hot Topic. Operat. Syst. (HOTOS-IX)*, 2003.

Dimitrios Serpanos is a Director of the Industrial Systems Institute, ATHENA and the Professor of Electrical and Computer Engineering with the University of Patras, Patras, Greece. His research interests include cyber–physical systems, embedded computing, and computer architecture. Serpanos has a PhD in Computer Science from Princeton University, Princeton, NJ, USA. He is a Senior Member of the IEEE and Member of the ACM, AAAS, and NYAS.

Muhammad Taimoor Khan is with the Institute of Informatics, Alpen-Adria University Klagenfurt, Klagenfurt, Austria. His research interests include the application of formal methods to reliability and security assurance of software systems, such as industrial control systems, computer mathematics-based systems and several Apache projects. Khan has a PhD from the Research Institute for Symbolic Computation, Johannes Kepler University, Linz, Austria.

Howard Shrobe is a Principal Research Scientist with MIT's Computer Science and Artificial Intelligence Laboratory (CSAIL), Cambridge, MA, USA. His research interests include AI, Cyber Security (particularly of control systems) and new computer architectures for inherently secure computing. Shrobe has a PhD from MIT. Since 1978, he has been a Member of the Staff of the MIT Artificial Intelligence Laboratory and its successors, CSAIL.

■ Direct questions and comments about this article to Dimitrios Serpanos, Industrial Systems Institute, ATHENA, PSP Building, Stadiou Street. GR-26504 Platani-Patras, Greece; e-mail: serpanos@isi.gr.