# Challenges in Secure Engineering of Critical Infrastructure Systems

Sridhar Adepu*, Eunsuk Kang†, Aditya P. Mathur*
*iTrust Center for Research in Cyber Security, Singapore University of Technology and Design, Singapore
{adepu_sridhar}@mymail.sutd.edu.sg, {aditya_mathur}@sutd.edu.sg
†School of Computer Science, Carnegie Mellon University, USA
{eunsukk}@andrew.cmu.edu

***Abstract*—Modern critical infrastructure (CI), such as water supply, smart power grids, and transportation networks, face major security challenges that arise due to complex interactions between software and physical components as well as human operators. Such systems are an attractive target for attackers who intend to disrupt the safe, normal operation of CI by exploiting vulnerabilities in software components such as the supervisory control and data acquisition (SCADA) workstations and programmable logic controllers (PLCs). In this *reference paper*, we elaborate on problems and challenges learned from our own experience in automating security analysis, assessment, and defense mechanisms for CI. These challenges are presented in the context of two real-world CI systems–namely, a water treatment plant and a water distribution system.**

***Index Terms*—Critical infrastructure protection, cyber physical systems, security automation, safety.**

## I. Introduction

Critical infrastructure (CI) systems such as power grids, transportation networks, and water supply systems, are considered vital to a nation's economy and prosperity. As CI systems become more capable and interconnected, these systems face a constant stream of threats from malicious actors, a single disruption in these infrastructure can have significant impact on safety and economy. For example, the 2003 blackout in US and Canada [1] was caused to a large extent by the failure that initially occurred in the communication system. Compared to traditional information systems, there are new major challenges to securing CI that arise from their unique system characteristics. In particular, CI is a type of cyber-physical system (CPS), where software components are coupled with physical processes in order to achieve real time monitoring and control. This coupling, however, also makes it possible to damage the physical system through cyber components, and vice-versa. New secure development methods that explicitly take into account interactions between cyber and physical components are needed.

The goal of this paper is two-fold: (1) to characterize major challenges in securing CI based on our own experience working with realistic testbeds for a water supply system (Section II) and (2) based on these challenges, to propose a set of future research directions on developing tools and methods for secure engineering of CI systems (Section III). In particular, securing CI will involve activities throughout an entire system life-cycle (from design to deployment and maintenance), and this paper proposes some of the ways in which software engineering methods (e.g., requirements analysis, modeling, verification, and usability) could play an important role in secure CI development.

## II. Testbeds

### A. System Characteristics

We introduce two testbeds that have been developed at the iTrust Center for the purpose of research on cybersecurity: the Secure Water Treatment (SWaT) and Water Distribution (WADI) testbeds. A key distinguishing feature of these testbeds is realism: Each testbed is a fully functional plant with all the elements of a full-blown industrial plant. The only key differences are that an industrial plant would have multiple paths for increased capacity, and physically larger versions of the individual components.

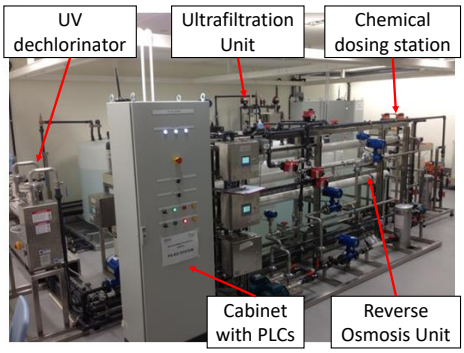


Fig. 1. The Secure Water Treatment (SWaT) testbed

**Secure Water Treatment (SWaT).** The SWAT testbed [2], shown in Fig. 1, is a room-size water treatment plant that performs various types of chemical processing to purify raw water. In total, there are six different stages in the water treatment process (pre-treament storage, dechlorination, reverse osmosis, etc.,). For the purpose of our discussion, the details of each stage is not crucial. Instead, we focus on the general characteristics of the system that are relevant to security.

SWaT can be regarded as a conventional industrial control system (ICS), in that its components can be classified into four types: (1) physical processes, such as water tanks or pumps, (2) sensors, which monitor the state of the physical processes (e.g., the level of water in a tank), (3) actuators, which directly

978-1-7281-4136-7/19/$31.00 ©2019 IEEE
DOI 10.1109/ASEW.2019.00030

interface with the physical processes to manipulate their state (e.g., activate a pump to pump the water out of a tank), and (4) controllers (typically implemented as a programmable logic controller (PLC)), which periodically read the sensor output and issue appropriate actuator commands to manipulate the physical processes as needed. For each of the six stages, SWaT contains a corresponding set of sensors, actuators, a PLC, and one or more physical processes (e.g., a raw water tank and a pump in Stage 1).

SWaT is also fitted with a supervisory control and data acquisition (SCADA) workstation that is used by system operators to monitor the status of the physical processes and manually override the actuators if necessary (e.g., if a water tank is about to overflow, turn off a valve to stop the flow of water into the tank). The SCADA system and PLCs communicate to each other through a local wireless network; in addition, the workstation is connected to the Web to enable remote operator access. These cyber connections expose the system to a wide range of security attacks. For instance, a remote attacker may use social engineering exploits or vulnerabilities in the workstation OS to compromise the supervisory software; an attacker with close proximity to the plant may perform man-in-the-middle attacks on the wireless network to modify sensor outputs or send fake actuator commands. In addition, a malicious insider with access to the plant may carry out various physical attacks (e.g., contaminate stored water).

**Water Distribution (WADI).** WADI [3] is a scaled down version of a real water distribution system that may be deployed in a city. The interesting part of this system, from the security perspective, is its interaction with SWaT. The water that is treated by SWaT is directly fed into WADI, which then emulates the distribution of water to households. This connection allows us to test the impact of cascading attacks across these systems (i.e., how does a cyber attack on one of the treatment stages in SWaT impact the rate of water supply?). Due to limited space, we omit other details of this system.

### *B. Security Challenges*

Assuming a model of an attacker whose goal is to force the testbeds into an unsafe state (e.g., manipulate the water tank into overflowing) and cause disruption to water supply, researchers at iTrust have developed several countermeasures for SWaT and WADI. In this section, we describe some of the attack detection and defense mechanisms investigated so far, and based on our experience, list major challenges in securing CI like SWaT and WADI.

**Threat Modeling and Vulnerability Detection.** Various attack models [4] for CPS were developed and demonstrated on SWaT and WADI. Based on the attack models, attacks were manually designed, launched and their impact observed. One common approach to evaluate the security of CI, and identify potential vulnerabilities, is to use existing attack benchmarks and datasets, as having been made available by researchers for different CPS testbeds [5] and used in the evaluation of different countermeasures [6]. Currently, however, these benchmarks are constructed through a time-consuming and error-prone manual process. This raises the following research challenges:***Research Challenge (RC) 1:*** *How to automate the construction and validation of attacks for in CI?* ***RC2:*** *How to develop a general benchmark of attacks that are reusable across CI?*

**Defense Mechanisms.** Various methods for detecting an attack have been investigated on SWaT and WADI. Ghaeini et al. [7], for example, monitor the network traffic with a hierarchical intrusion detection system, and Ahmed et al. [8] detect attacks by fingerprinting sensor and process noise. Other approaches learn models from physical data logs, and use them to evaluate whether or not the current state represents normal behaviour or not; some (e.g. [9], [10]) use unsupervised learning to construct these models, while Chen et al. [11] use supervised learning by automatically seeding faults in the control programs (of a high-fidelity simulator). Adepu et al. [12], [13] systematically and manually derive a comprehensive set of physics-based invariants and other conditions that relate the states of actuators and sensors. Although these methods have shown some success in detecting attacks, they do not provide rigorous guarantees or precise characterization of the types of attacks that they are able to detect. The following research challenges remain: ***RC3:*** *How to provide verifiable guarantees about classes of attacks being detected?* ***RC4:*** *How to monitor vulnerable physical processes and enforce actuator commands to prevent a transition into an unsafe state?*

**Incident Response.** When the defence monitors detects an attack, a simple and naive countermeasure is to shutdown the system. However, this response is not always desirable, since a typical CPS performs many critical functions that must be made continually available to its customers (e.g., water or power supply). Instead, a robust CI must be designed with capabilities to (1) respond to an on-going attack by performing actions to disable the attacker's access to the system and (2) recover from an successful attack by performing actions to move the system from an unsafe (e.g., water tank is about to overflow) to a safe state (all physical processes stable). At present such technology is not available, and the following research challenges remain: ***RC5:*** *How to respond to an on-going attack on-the-fly, by performing actions to disable the attacker's access to the system?* ***RC6:*** *How to recover from a successful attack by performing actions to move the system from an unsafe to safe state?*

## III. Research Directions

We envision that securing CI will require improvements in activities throughout an entire development lifecycle, from requirements analysis to testing and deployment. Based on our experience with the testbeds and the above challenges, we propose future research directions towards an effective methodology for secure development of CI.

**Integrating Safety and Security for CI.** Computer security, in general, deals with the protection of data and services. Safety, on the other hand, addresses the problem of preventing harm to the users or environment of a system. In the context of CI, the potential impact of a vulnerability is no longer

limited to data exposure or service takedown; it may result in a safety disaster. Consider, for example, the recent series of ransomware attacks carried out on hospitals around the world[1]. Although the loss of patient records is an undesirable outcome, the arguably greater risk is that the hospital staff would not be able to carry out medical procedures due to the unavailability of computers, possibly leading to worsening of patient conditions or even deaths. As CIs are increasingly being connected to the Web, identifying and mitigating undesirable interactions between safety and security is becoming more imperative.

Several researchers [14]–[16] have studied the convergence of safety and security concerns and identified the interdependencies between them. However, we believe that there are opportunities for new tools and techniques to address unique challenges in the context of CI safety and security. First, an automated requirements analysis technique that combines safety methods (such as hazard analysis or fault tree analysis [17]) from those in the security domain (e.g., threat modeling and attack surface identification [18], [19]) would be valuable for an engineer to discover how security exploits could lead to safety violations. Second, traditional safety-critical systems (like CI) are equipped with built-in monitors that are designed to detect random failures in physical components (e.g., sensors), but not intrusion from attackers; methods for developing and strategically placing a combination of safety and security monitors throughout the system would also be highly valuable. Finally, another promising direction is to devise architectural tactics or patterns to structure the system with a small core of trusted components so that a vulnerability outside the core does not result in safety violations [20].

**System-level Analysis.** A typical CI system is built as a collection of heterogenous components, including software (e.g., PLC and communication modules), hardware (sensors and actuators), physical processes (water tanks and chemical dispensers), and human agents (plant operators). A security compromise can occur in any one of these components, and understanding the impact of a vulnerability on the overall system will involve reasoning about interactions among these components. Performing this type of system-level analysis is particularly challenging in part because the characteristics of these components may be very different in nature (e.g., discrete for software versus continuous for physical processes). Further research is needed on (1) methods for specifying and composing models of these heterogenous components and (2) analysis techniques that leverage these models for identifying the attack surface and evaluating the impact of a vulnerability. For example, building on our preliminary work [21], we are developing a security modeling framework that combines a discrete model of the system architecture and controllers (specified in the Alloy language [22]) with a continuous model of physical processes (specified as timed automata in UPPAAL [23]) to automate the generation of attacks on SWaT.

A major challenge that lies beyond heterogeneous component interactions is achieving security for a *system of systems*. A single CI system is often deployed as part of a larger, more complex societal system with overarching requirements. For instance, SWaT and WADI together mimic the critical function of delivering clean water to civilians; a failure in either one of these systems may undermine this requirement and potentially lead to a catastrophic outcome. The system boundary, however, does not stop here; a smart grid system plays an important role in supplying power to both SWaT and WADI in order to ensure their continual operation. As systems become more interconnected and dependent on each other, the impact of a security attack on one of these systems is potentially amplified across system boundaries. To enable end-to-end analysis of system interactions, a new type of security analysis framework is needed; such a framework would support (1) abstractions for specifying and composing multiple systems, (2) analysis of the impact and propagation of a security failure across system boundaries, and (3) evaluation of potential mitigation and incident response strategies.

**Automation.** There has been a steady progress of techniques in formal verification, program analysis, and testing for detecting vulnerabilities in software. Relatively little work, however, has been done on developing similar types of methods for *responding* to and *recovering* from an attack; i.e., detecting when an attack has taken place, and performing appropriate actions to return the system to a desirable state. Currently, incident response in a typical CI is performed manually by a human operator, but this is problematic since the operator may not be able to react quickly enough to prevent the system from experiencing a catastrophic failure (e.g., irreversible damage to a physical process). An approach for (semi-)automating this task by, at run-time, detecting the presence and extent of an attack and synthesizing an appropriate response would be beneficial. For SWaT, we are investigating an approach that uses physical *invariants* to monitor the system for potential attacks (e.g., if the pump is activated, the level of water in the tank should decrease over time; a behavior that deviates from this may point to a compromised sensor) [6], [12]. The invariant-based monitor would produce information about which of the sensors might have been compromised, after which an automated response engine would perform actions to enable secondary sensors and generate actuator commands to bring the water level to a safe threshold.

Another challenging task that can benefit from increased automation is the construction of system and threat models. In systems like CI and CPS that rely on physical components, constructing a faithful, detailed model of physical processes (e.g., how the properties of water change depending on the amount of chemicals dispensed) can be particularly challenging even for domain experts [24]. One promising approach is to leverage recent advances in machine learning techniques to infer a model from observation logs. For example, researchers at the iTrust Center have explored both supervised [11] and unsupervised [9], [10] machine learning techniques to learn the physical models in SWaT.

**Human Factors and Usability.** Even though many secure

[1] https://www.theguardian.com/technology/2017/may/12/global-cyber-attack-ransomware-nsa-uk-nhs

development activities can benefit from improved automation, we expect that human agents will continue to make key security decisions in CI for the foreseeable future. In SWaT, a system operator is responsible for ensuring the safe operation of the plant by monitoring various sensors and performing appropriate actions in case of an anomalous system behavior, through the HMI and the SCADA workstation. Many SCADA systems, traditionally isolated to a local network, are now being connected to the Web. Although this increased connectivity has its benefits (e.g., allow remote operation), it also exposes the system to a wide range of security threats. Unfortunately, humans are often the weakest link in security, and there has been a number of successful attacks on SCADA that leverage social engineering [25]. Much work is needed on (1) improving conventional HMIs (which are safety-oriented) with security mechanisms (e.g., access control policies and minimum privilege enforcement) and (2) training system operators to raise their security awareness and react appropriately against potential social engineering attacks.

**Security tools.** Another avenue of research is to design security tools that are more usable and approachable to developers and operators of CI. To prevent the man-in-the-middle attacks on sensors and actuators, for example, data transmitted over the local network could be augmented with message authentication codes (MACs). Implementing this security mechanism involves configuring and deploying various network devices and application code with proper security settings. Even for those who are security experts, this is a complex, error-prone task that is often carried out in an ad hoc manner [26]. Usable tools for deploying, testing, and managing security configurations will play an important role in securing CI.

## IV. Conclusion

While notable progress has been made in defending CI against cyber-enabled attacks, there remain gaps in methods and technologies needed to fully realize the goal of robust and reliable defense. In this reference paper, we have identified such gaps and formulated open questions that could, in particular, benefit from software engineering methods, including requirements specification, architecture modeling, verification, reverse engineering, and usability.

**Acknowledgments.** This work was supported in part by the NSF award CNS-1801546, and by the National Research Foundation (NRF), Prime Minister's Office, Singapore, under its National Cybersecurity R&D Programme (Award No. NRF2014-NCR-NCR001-040) and administered by the National Cybersecurity R&D Directorate.

## References

[1] T. F. U.S.-Canada, Power System Outage, "Final report on the august 14, 2003 blackout in the united states and canada: Causes and recommendations," 04/2004 2004.

[2] A. P. Mathur and N. O. Tippenhauer, "SWaT: A water treatment testbed for research and training on ICS security," in *International Workshop on CySWater*. USA: IEEE, April 2016, pp. 31–36.

[3] C. M. Ahmed, V. R. Palleti, and A. Mathur, "WADI: A water distribution testbed for research in the design of secure cyber physical systems," in *The 3rd CySWater*, April 2017.

[4] S. Adepu and A. Mathur, "Generalized attacker and attack models for cyber physical systems," in *COMPSAC, 2016 IEEE 40th Annual*, vol. 1. IEEE, 2016, pp. 283–292.

[5] J. Goh, S. Adepu, K. N. Junejo, and A. Mathur, "A dataset to support research in the design of secure water treatment systems," in *International Conference on Critical Information Infrastructures Security*. Springer, 2016, pp. 88–99.

[6] S. Adepu and A. Mathur, "Assessing the effectiveness of attack detection at a hackfest on industrial control systems," *IEEE Transactions on Sustainable Computing*, 2018.

[7] H. R. Ghaeini and N. O. Tippenhauer, "Hamids: Hierarchical monitoring intrusion detection system for industrial control systems," in *Proceedings of the 2nd ACM Workshop on Cyber-Physical Systems Security and Privacy*. ACM, 2016, pp. 103–111.

[8] C. M. Ahmed, J. Zhou, and A. P. Mathur, "Noise matters: Using sensor and process noise fingerprint to detect stealthy cyber attacks and authenticate sensors in cps," in *Proceedings of the 34th ACSAC*. ACM, 2018, pp. 566–581.

[9] J. Goh, S. Adepu, M. Tan, and Z. S. Lee, "Anomaly detection in cyber physical systems using recurrent neural networks," in *18th Intern. Symp. on High Assurance Systems Engg.* IEEE, 2017, pp. 140–145.

[10] Q. Lin, S. Adepu, S. Verwer, and A. Mathur, "Tabor: A graphical model-based approach for anomaly detection in industrial control systems," in *Procd. of the AsiaCCS*. ACM, 2018, pp. 525–536.

[11] Y. Chen, C. M. Poskitt, and J. Sun, "Learning from mutants: Using code mutation to learn and monitor invariants of a cyber-physical system," in *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2018, pp. 648–660.

[12] S. Adepu and A. Mathur, "Distributed attack detection in a water treatment plant: method and case study," *IEEE Transactions on Dependable and Secure Computing*, 2018.

[13] ——, "Distributed detection of single-stage multipoint cyber attacks in a water treatment plant," in *Proceedings of the 11th ACM on AsiaCCS*. ACM, 2016, pp. 449–460.

[14] S. Kriaa, L. Pietre-Cambacedes, M. Bouissou, and Y. Halgand, "A survey of approaches combining safety and security for industrial control systems," *Reliability Engineering & System Safety*, vol. 139, pp. 156–178, July 2015.

[15] G. Sabaliauskaite and S. Adepu, "Integrating six-step model with information flow diagrams for comprehensive analysis of cyber-physical system safety and security," in *18th IEEE International Symposium on High Assurance Systems Engineering*, 2017.

[16] E. Lisova, I. Sljivo, and A. Causevic, "Safety and security co-analyses: A systematic literature review," *IEEE Systems Journal*, 2018.

[17] N. G. Leveson, *Safeware - system safety and computers: a guide to preventing accidents and losses caused by technology*. Addison-Wesley, 1995.

[18] A. Shostack, *Threat modeling: Designing for security*. John Wiley & Sons, 2014.

[19] G. McGraw, *Software security: building security in*. Addison-Wesley Professional, 2006.

[20] E. Kang and D. Jackson, "Dependability arguments with trusted bases," in *IEEE Intl. Conf. on Requirements Engineering*, 2010, pp. 262–271.

[21] E. Kang, S. Adepu, D. Jackson, and A. P. Mathur, "Model-based security analysis of a water treatment system," in *In Proceedings of 2nd International Workshop on Software Engineering for Smart Cyber-Physical Systems (SEsCPS'16)*, May 2016.

[22] D. Jackson, *Software Abstractions: logic, language, and analysis*. MIT Press, 2012.

[23] J. Bengtsson, K. G. Larsen, F. Larsson, P. Pettersson, and W. Yi, "UPPAAL - a tool suite for automatic verification of real-time systems," in *Hybrid Systems III: Verification and Control, Proceedings of the DIMACS/SYCON Workshop on Verification and Control of Hybrid Systems*, 1995, pp. 232–243.

[24] I. Ruchkin, S. Samuel, B. Schmerl, A. Rico, and D. Garlan, "Challenges in physical modeling for adaptation of cyber-physical systems," in *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*. IEEE, 2016, pp. 210–215.

[25] A. Matrosov, E. Rodionov, D. Harley, and J. Malcho, "Stuxnet under the microscope," *ESET LLC*, September 2010.

[26] S. M. Bellovin and R. Bush, "Configuration management and security," *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 3, pp. 268–274, 2009.