Leveraging Software-Defined Networking for Incident Response in Industrial Control Systems

Andrés F. Murillo Piedrahita, Universidad de los Andes

Vikram Gaur, Jairo Giraldo, and Álvaro A. Cárdenas, University of Texas at Dallas

Sandra Julieta Rueda, Universidad de los Andes

// Software-defined networks (SDNs) and network function virtualization (NFV) can facilitate automatic incident response to a variety of attacks against industrial networks. The combination of SDNs and NFV show promise in providing novel defense-in-depth solutions for industrial systems. //

MANY INDUSTRIES AND critical infrastructures are monitored and controlled by industrial control systems (ICSs). These systems control power grids, water and wastewater management, oil systems, and manufacturing, among others. Because most of these systems are safety-critical, any potential cyberattack may cause significant physical as well as economic damage. Recent attacks on ICSs such as the power-grid attack in Ukraine[1] or the recent hackings of US nuclear plants[2] show the dire need for improving the security of Internet-connected ICSs. Although several mechanisms to detect attacks have been developed, there is very little work on how to automatically respond to these alerts; most responses are manual or are hard-wired with a fixed response that cannot be configured.

In this article we show how *software-defined networks* (SDNs) and *network function virtualization* (NFV) technologies can help us design automatic incident-response mechanisms for ICSs. This kind of infrastructure enables the implementation of a variety of automatic reactions.

In particular, this article focuses on how to respond to attacks. We assume there is an intrusion detection system (IDS) that raises alerts indicating that specific devices have been compromised, and our goal is to design an incident-response policy to reconfigure the network and mitigate the impact of the attack. Designing and testing IDS effectiveness for ICSs is a different problem, necessary to approach but orthogonal to our effort here. The reader interested in learning more about IDSs for ICSs can find more details in our previous work,[3] which also includes a survey of a variety of techniques for detecting attacks.

0740-7459/18/$33.00 © 2018 IEEE

The rest of the article is organized as follows. We first present a high-level overview of the new opportunities software-defined infrastructures have for providing flexible and reliable incident-response mechanisms. We then present a prototype, as a proof of concept, to show how a software-defined infrastructure can be used to automatically respond to intrusions and keep the system in safe operation while sustaining attacks.

Software-Defined Infrastructures for ICSs

SDNs and NFV are technologies that improve network capabilities by enabling better management of network traffic flows, network visibility, and the deployment and control of network functions using software instead of hardware-specific middleboxes.

SDNs enable network reconfiguration and rerouting using a programmatic approach. The advantages of SDNs have already been recognized in traditional information technology networks[4] and are beginning to be explored in cyber-physical systems.[5,6] These advantages originate from two key points of the SDN paradigm:

- the development of software in COTS generic servers (as opposed to the traditional use of proprietary hardware and middleboxes with limited software programmability) and
- the separation between control (how routers discover and maintain routing paths) and data planes (how packets are forwarded between devices) in telecommunications.

In addition, networks do far more than forward packets; they process traffic through network functions like proxies, firewalls, intrusion protection systems, and so on. These functions traditionally have been implemented in middleboxes—dedicated hardware devices inspecting, filtering, or manipulating network traffic. This paradigm has serious disadvantages such as high capital investment due to costly middleboxes, the difficulty of deploying new services because of the need to reprogram or reconnect these middleboxes, and the inability to scale services to variable demand. NFV is a new telecommunications paradigm that enables the implementation of these network functions using software, virtualization, and general computing equipment rather than dedicated hardware.

SDNs and NFV have been proposed and used in general IT networks, but they have not yet been adopted by the ICS community. Nevertheless, industrial systems are evolving to provide better efficiency, availability, and security, due to advances in communications and embedded systems. In particular, in the context of Industry 4.0, SDNs can provide ICSs with the required global vision and flexibility to interconnect legacy telecommunication protocols, simplify detection of suspicious traffic, and implement dynamic security policies.[7]

Given this modernization of industrial systems, in this article we study how SDNs and NFV together can bring programmability to ICSs, enabling the development of new attack–response solutions.

The Benefits of SDNs and NFV for ICSs

In ICSs, a physical process is controlled by a set of *programmable logic controllers* (PLCs). PLCs use sensors to gather information about the process state and send commands to actuators in order to change process behavior, while a *supervisory control and data acquisition* (SCADA) center collects and presents the gathered information to human operators. The network through which SCADA servers communicate with remote controllers or terminal units is traditionally called a *supervisory network*.

While PLCs traditionally interact with sensors and actuators in the field through analog interfaces (4 to 20 mA), the growing number of sensors and actuators and their increased capabilities have resulted in new *field networks* where the PLCs interface with remote I/O boxes using new Ethernet-based industrial protocols.

SDNs and NFV can be integrated with industrial control networks at both the supervisory and field networks. Another advantage of SDNs is that they enable us to deploy different defenses without modifying legacy systems, industrial network protocols, or the logic behavior of traditional ICS elements.

In particular, SDNs and NFV make it possible to run components that implement different detection techniques and to start different defense techniques, dynamically scaling VNF (virtual network function) resources up or down, according to different attack scenarios. Potential incident–response use cases include the following:

- To mitigate an attack, the SDN controller can reroute the attacker's traffic to an ICS honeypot. This way, administrators can keep an attack active to obtain more information about the attacker's targets and methods, while at the same time isolating and protecting the real process. The ICS honeypot needs to provide fine-grained emulation of the devices and the physical process under control in order to deceive the attacker.

- To mitigate an attack on the sensors, the software-defined infrastructure can change the source of the readings given to the PLCs. The SDN controller can change forwarding rules to drop communications from a compromised sensor, temporarily instantiate a simulator of the physical process to generate estimated values, and give the values from the simulator to the PLCs. Nevertheless, this solution must be temporary. Using a simulation instead of sensor readings for extended periods of time will create a system with open-loop control (instead of closed-loop control), which is a long-term problem.
- The system may also transfer services from compromised devices to redundant devices. This approach traditionally has been used in the fault-tolerance community, but SDNs and NFV do not require investing a priori on capital equipment; instead, we can use virtualized resources.

The next section presents a prototype we developed as a proof of concept to show the feasibility of these defenses.

Software-Defined Infrastructures for ICS

In this section we describe a detailed implementation of SDN-enabled automatic response to an attack against an ICS process.

We consider a water treatment process[8] with three PLCs: PLC 101, PLC 201, and PLC 301. PLC 101 and PLC 301 control the water level in two tanks: Tank 101 and Tank 301. PLC 201 monitors a pipe that connects both tanks and measures water pH to activate (ON/OFF) a chemical-dosing pump to maintain the quality of the water within desirable limits. PLC 101 controls Tank 101 inlet and outlet flows by reading levels from sensor LIT 101 and sending ON/OFF commands to valve MV 101 and pump P 101. Similarly, PLC 301 controls Tank 301 through sensor LIT 301 and pumps P 101 and P 301.

A Cosimulation

To simulate the physical behavior of this process and emulate network components of this system, we extended MiniCPS[9] to provide SDN functionalities for both the supervisory and field networks. MiniCPS extends Mininet,[10] a light virtualization environment tailored for SDN experiments and emulation, to support Ethernet/IP, an industrial network protocol commonly used in ICSs for communications between the supervisory and field networks. MiniCPS also models communications between PLCs; however, it does not implement the field network (communications between PLCs and sensors and actuators).

We extended MiniCPS to be able to model a field network; our extension allows PLCs to interact with the physical process through communication with sensors and actuators. We also added virtual enabled functions, like detection and incident-response solutions that MiniCPS does not consider. Also, the original MiniCPS library implements Ethernet/IP using blocking sockets, so each host can interact only with another host at the same time. We modified the implementation to use asynchronous sockets to be able to send nonblocking messages; PLCs can now receive messages in a nonblocking way and concurrently send control actions to various actuators.

Figure 1 shows our extended topology in which PLCs communicate with sensors and actuators. It has a full topology of the water system, with three control loops in the field network.

Each control loop is represented by a LAN, while sensors, actuators, PLCs, and IDSs are represented by hosts. A Mininet switch connects each LAN, and a host acting as a router connects the three switches. An SDN controller manages the network, and a SCADA system periodically receives reports about the system state.

Our scenario and our SDN application are available on GitHub. (The environment is available at github.com/Cyphysecurity/ICS-SDN.)

An SDN Controller Application

We extended the POX SDN controller to handle a topology map of the ICS elements. Figure 1b illustrates our incident-response system:

1. The IDS instance always receives a copy of the information reported by the sensor LIT 101. The IDS[3] uses a mathematical model to identify differences between sensor readings and estimated values. If the difference is greater than a given threshold, the IDS notifies the SDN controller.
2. The SDN controller follows a preconfigured incident-response policy for the reported attack, like discarding packets from the compromised sensor and replacing them with the estimated

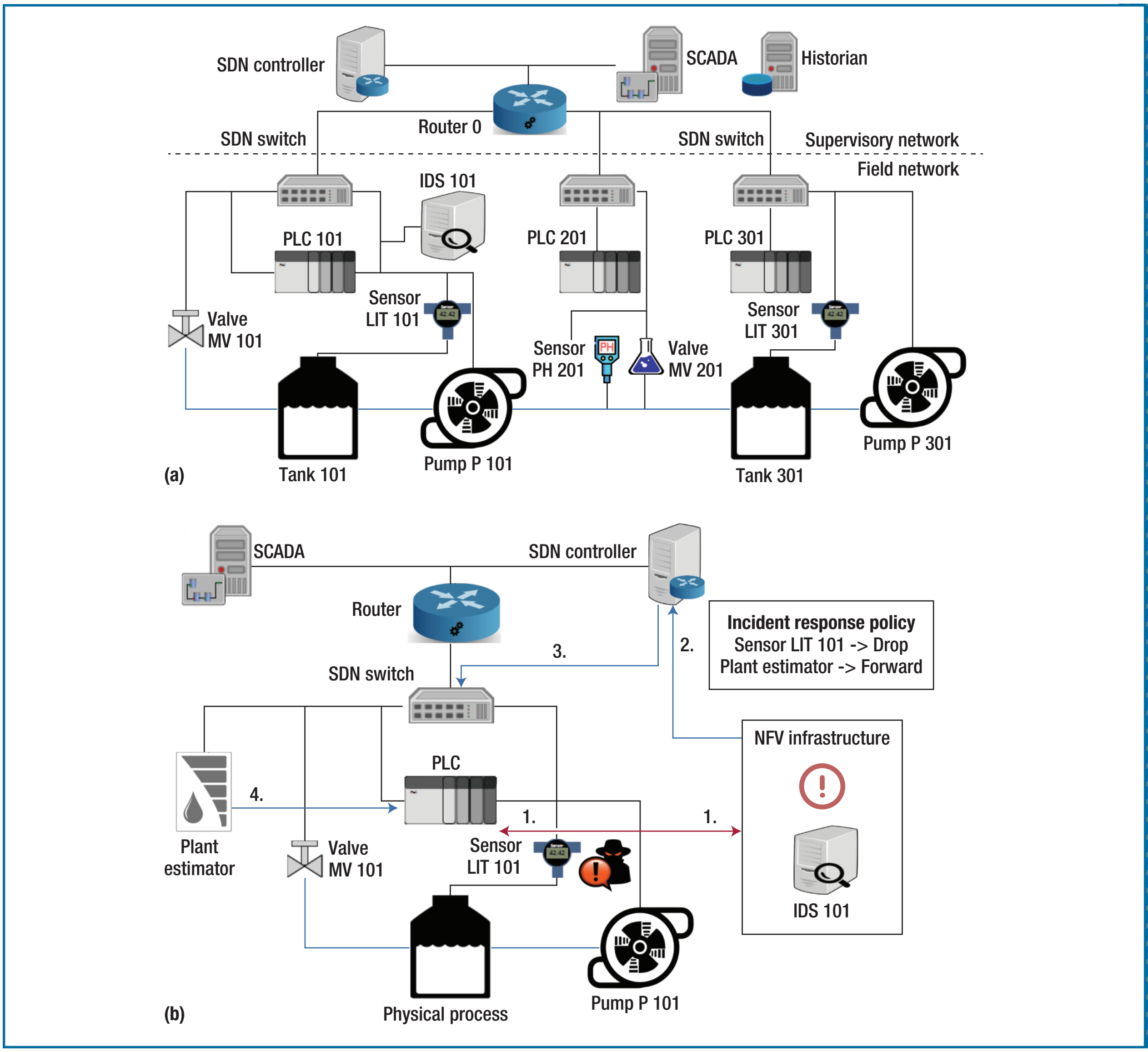

FIGURE 1. We implemented a prototype to enhance industrial control system (ICS) field networks with a software-defined network (SDN). (a) An enhanced industrial-control-network topology. The SDN switch is used by programmable logic controllers (PLCs), sensors, and actuators, as well as the supervisory control and data acquisition (SCADA) servers. The SDN controller can perform automatic incident-response configuration. (b) An enhanced ICS with an SDN. The intrusion detection system (IDS) notifies the SDN controller when it detects an attack on the sensor. The controller creates a flow entry in the switch to drop packets from the compromised sensor and forward the estimations from the plant estimator.

values, which in this case are also computed by the IDS.

3. The SDN controller modifies the flow table of the SDN switch to perform the configured response.
4. The plant is controlled by the estimated values.

Although we chose a particular IDS,[3] our software-defined infrastructure supports other detection approaches to alert the SDN controller. The alternative source of trusted values may also be different; however, our approach requires a

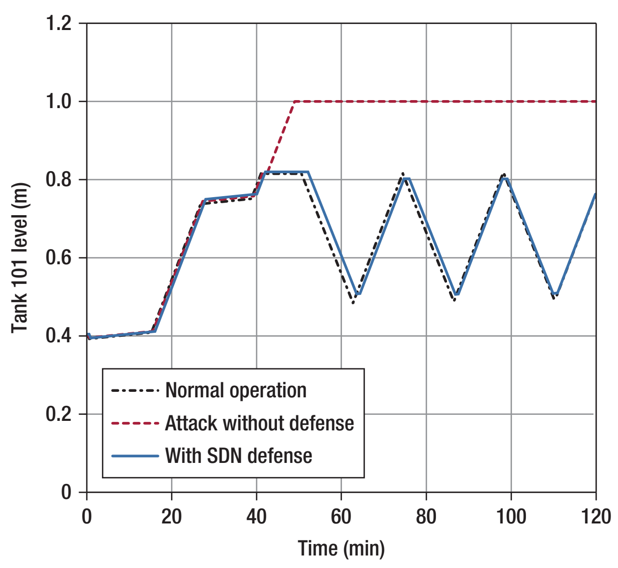

FIGURE 2. An attack scenario with a compromised sensor. Approximately 20 minutes after operation, sensor LIT 101 is compromised and starts sending a lower level value. Without a defense, PLC 101 never activates pump P101, and Tank 101 overflows. When the defense is present, the IDS identifies a malicious behavior and notifies the SDN controller to drop packets from the sensor. The system uses estimated values, and the plant behaves close to normal.

high-fidelity simulation of the controlled process to generate these values.

We consider that a software-defined infrastructure could greatly help increase ICS system resiliency because the logic of the ICS elements does not need to be modified, and SDN handles the dynamic reconfiguration of the network. Although our application is simple, this model can be extended to include different detection mechanisms and various incident-response policies for different types of attacks.

Experiments

The system must keep the water level of the two tanks between 0.5 and 0.8 of their capacity. Our experiment tested whether the system detects and reacts to an attack that compromises sensor LIT 101 to always report a water level of 0.3 below the real value. If that attack is successful, PLC 101 will never see a water level above 0.5 and will keep valve MV 101 always open and pump P 101 always closed; Tank 101 will overflow, and Tank 301 will never be filled.

Figure 2 shows the results of the experiment. When Tank 101's level reaches 0.5, the malicious sensor starts sending wrong readings. The IDS instance detects such behavior and notifies the SDN controller. Then, the SDN controller creates flow entries in the SDN switches to start dropping the readings from the malicious sensor and start forwarding the estimated values. The figure shows that the behavior of the plant under attack is similar to the scenario with no attack, indicating that the SDN-based incident-response system increases the resiliency of the system.

To quantify the response time of our system, we measured the time it takes the SDN controller to modify the flow table at the SDN switch, in order to complete the rerouting procedure. Specifically, we measured the time between the generation of the alert at the IDS and having the switch flow table modified to reroute traffic. We ran this experiment 10 times. The result was 2.3 ± 0.24 ms. Although this result was obtained in an emulation environment, we argue that it does not fall too far from a real environment setup. The only delays that our emulation did not consider were those created by network congestion and the medium access control mechanisms, such as those present in Ethernet. Nevertheless, in a real ICS, the network could be provided with enough resources to maintain the network latency within acceptable limits.

These delays are reasonable for physical systems with relatively slow dynamics, like the water tank in our example or the pH level changes from PLC 201. Water, gas, or oil extraction, treatment, and distribution involve many physical processes (in addition to several chemical processes) with slow dynamics. While we claim that our solution can be applied to a large set of these slow-dynamic ICSs, care must be taken in other environments like the power grid, where low latency is required. In future work we will explore the applicability of our success story in ICSs to power systems and evaluate the timing and delay constraints and requirements.

ABOUT THE AUTHORS

ANDRÉS F. MURILLO PIEDRAHITA is a doctoral student in engineering at Universidad de los Andes. His research interests are network security, software-defined networking, and network function virtualization. Murillo received an MSc in electrical engineering from Universidade Federal do Rio de Janeiro. Contact him at af.murillo225@uniandes.edu.co.

VIKRAM GAUR is a software development engineer at Amazon. His research interests include software-defined networking, network function virtualization, and cryptography. Gaur received a master's in computer science from the University of Texas at Dallas. Contact him at vikram.gaur@utdallas.edu.

JAIRO GIRALDO is a research associate in the Computer Science Department of the University of Texas at Dallas. His research interests include security and privacy in control systems, multiagent systems, and distributed control of the smart grid. Giraldo received a PhD in electrical engineering from Universidad de los Andes. Contact him at jairo.giraldo@utdallas.edu.

ÁLVARO A. CÁRDENAS is an assistant professor in the Department of Computer Science at the University of Texas at Dallas, where he's a member of the Cyber Security Research and Education Institute. His research interests include cyber-physical systems and IoT security and privacy, network intrusion detection, and wireless networks. Cardenas received a PhD in electrical and computer engineering from the University of Maryland, College Park. He has received an NSF CAREER award, best-paper awards from the IEEE Smart Grid Communications Conference and US Army Research Conference, and a fellowship from the University of Maryland. Contact him at alvaro.cardenas@utdallas.edu.

SANDRA JULIETA RUEDA is an assistant professor of systems and computer engineering at Universidad de los Andes. Her research interests are software system security, access control, policy analysis, and policy generation. Rueda received a PhD in computer science and engineering from Pennsylvania State University. Contact her at sarueda@uniandes.edu.co.

We have shown how SDNs and NFV may be used together to enhance ICS security. We implemented an open source prototype, as a proof of concept. It is available to researchers who want to test our implementation and extend our system as we described in the section "A Cosimulation." As far as we are aware, our software is the first implementation of SDN-enabled incident response to attacks that try to manipulate a process in ICS networks.

Acknowledgments

This work is partially supported by NIST awards 70NANB16H019 and 70NANB17H282, by the National Cybersecurity FFRDC, by the Air Force Office of Scientific Research under award FA9550-17-1-0135, and by the Colombian Administrative Department of Science, Technology, and Innovation (Colciencias).

References

1. R.M. Lee, M.J. Assante, and T. Conway, *Analysis of the Cyber Attack on the Ukrainian Power Grid*, tech. report, SANS Industrial Control Systems, 2016.
2. M. Riley, J.A. Dlouhy, and B. Gruley, "Russians Are Suspects in Nuclear Site Hackings, Sources Say," Bloomberg, 2017; www.bloomberg.com/news/articles/2017-07-07/russians-are-said-to-be-suspects-in-hacks-involving-nuclear-site.
3. D.I. Urbina et al., "Limiting the Impact of Stealthy Attacks on Industrial Control Systems," *Proc. 2016 ACM SIGSAC Conf. Computer and Communications Security* (CCS 16), 2016, pp. 1092–1105.
4. P. Sun et al., "A Network-State Management Service," *Proc. 2014 ACM Conf. SIGCOMM* (SIGCOMM 14), 2014, pp. 563–574.
5. X. Dong et al., "Software-Defined Networking for Smart Grid Resilience: Opportunities and Challenges," *Proc. 1st ACM Workshop Cyber-physical System Security*, 2015, pp. 61–68.
6. A. Aydeger, K. Akkaya, and A.S. Uluagac, "SDN-Based Resilience for Smart Grid Communications," *Proc. 2015 IEEE Conf. Network Function Virtualization and Software Defined Network* (NFV-SDN 15), 2015, pp. 31–33.
7. J. Wan et al., "Software-Defined Industrial Internet of Things in the Context of Industry 4.0," *IEEE Sensors J.*, vol. 16, no. 20, 2016, pp. 7373–7380.
8. A.P. Mathur and N.O. Tippenhauer, "SWaT: A Water Treatment Testbed for Research and Training on ICS Security," *Proc. 2016 Int'l Workshop Cyber-physical Systems for Smart Water Networks* (CySWater 16), 2016, pp. 31–36.
9. D. Antonioli and N.O. Tippenhauer, "MiniCPS: A Toolkit for Security Research on CPS Networks," *Proc. 1st ACM Workshop Cyber-physical Systems-Security and/or Privacy* (CPS-SPC 15), 2015, pp. 91–100.
10. B. Lantz, B. Heller, and N. McKeown, "A Network in a Laptop: Rapid Prototyping for Software-Defined Networks," *Proc. 9th ACM SIGCOMM Workshop Hot Topics in Networks* (Hotnets 10), 2010, pp. 19:1–19:6.