Monitoring Security of Networked Control Systems: It's the Physics

Chuck McParland | Lawrence Berkeley National Laboratory
Sean Peisert | Lawrence Berkeley National Laboratory and University of California, Davis
Anna Scaglione| University of California, Davis

Physical systems must be operated safely and correctly. One way of enhancing operational safety is by leveraging specification-based intrusion detection to monitor for physical constraint violations. This additional security layer enhances protection from both outsider attacks and insider mistakes.

Of today's many physical systems that impact or are impacted by networked computers, computerized control systems represent a class of applications of both great importance and national interest. A growing trend, motivated by economic and design purposes, is to embed intelligence into physical systems and connect the various systems through a common industrial control network whose dual purpose is to monitor and control the plant. These cyber-physical systems often rely on a supervisory control processor that allows operators to monitor the state of each physical device in the plant and change their operating set points. The design of protection equipment was originally based on hardware but is increasingly performed by transferring analog and digital values from small controllers embedded in the devices to a central plant control system, which receives sensor information and exerts commands through networks. As a result, there exists a situation in which certain commands sent over the network can manipulate physical devices. If the manipulation is done in a way that puts the cyber-physical system in an unsafe state, a simple cybercommand could result not only in damage to equipment, but also in injury or loss of life.

The Safe and Secure Operation of Physical Devices

In general, embedded systems used in industrial control networks include programmable logic controllers (PLCs), distributed control systems, and safety-instrumented systems. In the hierarchical Sensory Control and Data Acquisition (SCADA) reference model, remote terminal units refer to the plant controllers and intelligent electronic devices refer to the controllers embedded in the machines. Due to the presence of digital controllers that can turn parts of the system on and off, it's more appropriate to refer to a hybrid physical state than to a physical state, which typically refers only to the analog portion of the system and not to the state of its switches. Most protective controls modify the system's hybrid state—often when the analog states reach some boundary conditions—and the change impacts the underlying physical laws to which the hybrid state refers.

The transition from using direct serial links for each communication to using shared Ethernet media and the Internet protocol (IP) was a big leap for these systems. In the past, information networks and controls of this type were either in isolated environments—such as

1540-7993/14/$31.00 © 2014 IEEE

manufacturing plants and power generation facilities—or were limited to control infrastructures such as those used for electricity transmission or gas pipelines. The operation and management of these facilities were characterized by strict, continuous operator scrutiny and included formalized contingency planning based on safety and availability requirements. However, these networked systems are increasingly used in small plants with ad hoc network configurations and are opportunistically designed to lower cost and efficiently meet local needs. After their installation and initial commissioning, these systems rapidly become opaque automation tools with little documentation and even less local understanding of their low-level design.[1]

Whether in industrial, building, or manufacturing environments, systems are beginning to exploit cloud-based, "big data" analysis tools and provide networked control to more efficiently harness renewable distributed generation and electric energy from the grid. Even the rapid expansion of the home automation market is creating opportunities for cloud-based control and optimization of many residential subsystems, such as HVAC. The full impact of remote automated control is yet to be appreciated, as seen in Google's 2014 acquisition of Nest Labs. As the use of these network-based control systems grows, equipment operators and building occupants will expect that they're safely operated.

These physical systems have always been designed with extremely high degrees of safety in mind through a technique called *safety engineering*, which sets the requirements and best practices for how systems should be designed and managed by human operators, along with permissible failure scenarios. These principles have influenced designers of large computer systems that operate—and fail—in well-understood and controlled ways. However, unlike modern computer systems that are upgraded every four to six years, many cyber-physical systems are an amalgam of old and new components operating side by side, often with inconsistent operating controls, algorithms, and guidelines. Furthermore, these updated controls are local to the device environment, which means there's no mechanism or certification process in place that would ensure their safe operation from remote sites.

Much of the existing research relating to the security of these systems is based on the assumption that "cyber" and "physical" elements should be viewed in tight, separate compartments, in which the cyberinfrastructure or the physical devices under control can't be the root cause of failures or security violations in the other domain. This intersection of safety engineering and computer security has become one of the most significant sources of concern in the current analysis of cyber-physical systems. And, because these elements can be directly addressed by other devices in the network—due to the growing use of modern IT networking infrastructures—the flow of monitoring and control messages can be subverted in unanticipated ways. The design effort to harmonize a system's "appearance" to control algorithms also masks the fact that the system contains very different components. As a result, the roles of supervisory control elements in ensuring safe operations can become overly fuzzy and subject to compromise. Have designers of such systems made unwarranted and potentially conflicting assumptions about which particular discipline or set of systems— cybersystems and their designers, and physical systems and their designers—is responsible for the safe and secure operation of physical devices?

As the targets of the Stuxnet worm[2] and the wireless attack leading to the sewage spill at the Maroochy Water Station in Australia[3] now know, control systems often allow behavior that's damaging to individual devices. But attacks aren't necessarily intentional, as shown in the accidental destruction of the Sayano-Shushenskoe hydroelectric plant in Russia.[4] Each case is primarily a result of control systems being misinformed of equipment operation monitoring values or simply ignoring physical systems' operating tolerances while low-level serial protocols command devices to operate beyond their physical limits. These attacks all suggest that spoofing and misinformation of isolated control sites that lead the systems to insecure or emergency states can go unnoticed and cause catastrophic consequences. An even more insidious problem is the potential of well-coordinated local behaviors that can, when properly orchestrated at a system-wide level, achieve damaging results.[5]

Of course, key failures that systems must defend against aren't limited to external attacks but can also originate from operator errors and malicious activities by "privileged" users. In those cases, solutions that simply address access control are typically useless because authorized users, by definition, already have access.

When considering the ramifications of the potential for widespread damage to infrastructure and property, it becomes clear that both new and legacy systems must be secured in a more robust and clearly understood manner. To address this problem, our solution closely integrates the cybersecurity components with control commands, physical constraints, and safety constraints of physical devices to mitigate a substantial class of vulnerabilities in cyber-physical systems. In this article, we discuss our reasoning and suggest some ways in which this might be accomplished. To exemplify this general approach, we propose using intrusion detection systems (IDSs) that monitor cyber-physical systems for commands that cause those systems to exceed their physical limitations, similar to how IDSs typically monitor for commands that could put computer systems in an unsafe state.

Background and Related Work

Today's cyber-physical security research ranges from exposing the lack of basic computer security practices and policies in utility working environments to measuring and ensuring the integrity of data repositories used as part of utilities' increasingly networked operations. Research that highlights security gaps in utilities' networks often fits in a classical IT security framework and strives to be somewhat independent of applications.

Two very common control protocols are Modbus and DNP3. Both originally ran over dedicated lines, such as serial connections, and evolved to run over TCP. Modbus is extensively used in industrial applications due to its robustness, ease of deployment and maintenance, and ability to move raw data without placing many restrictions on vendors' designs. DNP3 is seen more often in power-related environments. As with most protocols used in control environments, including both vendor proprietary protocols as well as other "open" protocols, neither Modbus nor DNP3 originally had a security layer, and neither was designed for use in open networks. Security is traditionally achieved by limiting authorized access to the control software (passwords) or strictly administrating physical separation (firewalls and VPNs). However, anyone who can insert themselves or malware behind the firewall can issue commands to systems with potentially harmful results to the physical devices under control.

Various additional network security techniques were added over time, including authentication, encryption, and digital signatures as well as various types of network intrusion detection. Various commercial and research efforts in intrusion detection represent examples of techniques used to examine network traffic to and from control systems.[6] However, despite protection for authentication, integrity, and confidentiality of cyberassets, numerous counterexamples have indicated that these techniques can't guarantee security and safety and that standard network security techniques can't stop many classes of attacks designed to manipulate cyber-physical devices.[3] Key research has been performed to build safety models for cyber-physical and power systems,[7,8] but we believe that such safety models must also be integrated into the IT network control itself.

Merging Safety Engineering and Computer Security

Our approach can be seen as an extension of Calvin Ko and his colleagues' *specification-based intrusion detection*.[9] Specification-based intrusion detection, the opposite of misuse detection, defines a set of good properties and looks for behavior outside those properties. Although typically intended for traditional network and host-based intrusion detection, specification-based intrusion detection has also been applied to control programs.[10] This method is particularly suited to detecting attacks on cyber-physical systems because well-defined, specific physical processes are exactly what limit the operational bounds. The method has been effective in other research efforts focused largely on protocol violations[6] as well as in examining physical violations.

In our research, we characterize physical limitations using pre- and post-conditions of hybrid state transitions. We correlate the behavior of cyber-physical devices from local activity captured across multiple control devices that communicate through the same network by tapping multiple dataflows, which ensures that devices behave according to the valid security and safety specifications of the entire subsystem in the IDS's reach.

In our method, we stipulate that the IDS can maintain knowledge of the controlled devices' hybrid physical states in parallel with any existing safety systems. It can also monitor the sensor data exchanged on the network to update that state's information and monitor commands that would change the state. So, if a turbine is rotating at a particular velocity and a command is sent across the network to change that velocity, the IDS should observe the command and the set point to determine whether the command would cause damage to the physical system.

To do this, a basic first step is to analyze the consistency of the data and command flows among the various terminals using the embedded automation systems' hybrid control mission. This aspect functions well with both internal and external threats. For example, although an effective network IDS might prevent a malicious external attempt to invoke damaging commands, these same commands could be issued (possibly in error) by an authorized insider—in effect, bypassing the sophisticated perimeter protections that were put in place to block external attacks. The use of a common cybersecurity or control processing layer designed to monitor the command stream, regardless of the source, provides a consistent environment in which control commands and system status data can be vetted, logged, and passed on to control system elements for processing. As threat modalities evolve and appropriate attack filters are created, such a framework provides a common architectural point for additional capabilities. In addition, a security framework that's aware of all command streams entering a target system allows for the effective integration of model-based control and safety analysis techniques.

For example, suppose the communication system that connected a rotating AC generator to the network allowed specific commands to be sent because the physical limitations of the generator's operation weren't considered. If those limits were considered, the specific description of how fast and how often the generator can change rotational velocity should be represented as a mathematical equation and built into the communication mechanism. By monitoring the commands sent to the generator and what the generator is and has been doing, an intrusion prevention mechanism can use that data to look for commands that would cause the device to violate its limits. Thus, when a device is operating in such a way that it's in danger of self-destructing—violating its own protocol—the commands shouldn't be sent until the generator has returned to a safe state (defined by the protocol).

> **Pulll quote is approximately 20–25 words. Pulll quote is approximately 20–25 words. Pulll quote is approximately 20–25 words. Pulll quote is approximately 20–25 words.**

Assume now that the regular communication of the generator's rotating speed is spoofed and that the physical quantities the IDS reported and intercepted don't exceed the operating limits in the present hybrid state, while in reality the generator is spinning out of control. By integrating the direct information with, say, an electric meter reading from other parts of the electrical network, the IDS can detect the inconsistency in the flow of power to the system and, ultimately, detect the attack.

We don't believe that this "physics-based" intrusion prevention mechanism should necessarily be the sole protection mechanism, because this could lead to a single point of failure. Ideally the PLC, or some other embedded automation system, would have its own safety mechanism. However, the network intrusion prevention mechanism can support proper safety engineering techniques by duplicating some PLC functions and double-checking what the PLC could fail at enforcing.

Theoretically, any tool that can monitor and parse the body of control system network traffic can be adapted to monitor network traffic, looking for commands that alter the devices' physical states. As we mentioned, several commercial and open source IDSs are capable of parsing Modbus TCP and DNP3 packets. Despite the theoretical capabilities of such tools, they're rarely employed by actual industrial control system operators. One possible reason is the lack of successful, published demonstrations of the approach's effectiveness.

Developing Our Approach

We developed our method using the open source Bro network security monitor.[11] The Bro system provides a flexible framework that allows passive inspection of all network packets by a user-configurable set of program scripts. Depending on the design of these scripts, packets can be inspected for strict conformance to specific protocol guidelines (such as TCP/IP or Modbus) and can be parsed to reveal information about their originating processor. Bro can also be extended to allow user-defined scripts to inspect these packet streams and, in our case, analyze them in the context of the physical system with which they're interacting. Based on device protocols and physical limitations, we developed specifications and enhanced monitoring by using Bro to determine when a system is about to violate a protocol specification. Bro supports a highly "stateful" view of application layer behavior and is readily adaptable to new protocols and analyzers. Bro was recently augmented with native abilities for monitoring Modbus TCP and DNP3 traffic. In addition, Bro's client-side Broccoli (Bro Client Communications Library) capability enables not just passive monitoring but active probing of PLCs.

We implemented specification-based SCADA command analyzers directly in the Bro framework using physical constraint algorithms. To accomplish this, we added low-level analyzers that examine control system–specific protocol packets for both Modbus TCP and DNP3, along with higher-level analyzers that interpret device command and data streams in the context of each device's physical capabilities. These higher-level, context-sensitive analyzers inspect device-monitoring streams to verify that they're "physically meaningful" and verify the safe intent of device commands by examining the risk of future contingencies that might arise due to the packet. The challenge, in this case, is to strike the right compromise between risk and complexity of the assessment. Finally, we utilized Bro's stateful layer to represent and track physical systems' operational modes and implement synchronization semantics among cooperating network protocol and physical constraint analyzers.

To construct the specifications in the form of Bro scripts, we first abstract away the specific details of communication protocols and technologies to focus on the physics models of the devices being controlled. For example, when parsing a Modbus TCP packet, the communication aspect quickly identifies the PLC being communicated with; the memory address being read from or written to; and the meaning of the data contained in that event, such as revolutions per minute. For a "write" command, regardless of whether the value being written changes the set point for the device's RPMs to an absolute value or a change in value, the IDS's state can be updated with the command and act accordingly.

The goal of this modeling activity is to describe the action space for the networked system in a tractable way to derive the best policies from the defenders' perspective. These models are used to explore feasible attacks with the same system constraints. For example, if systems are networked so that there's a best "safe schedule" for electrical loads, attained by controlling loads in concert, the control system can be used to "optimize" for mayhem with the same physical and network constraints. So, the security question and the load control question are inextricably intertwined. This question is fundamentally different from the data attack problems that others have considered, making our approach stand out. The important feature of the attacks we're studying is that the networked system produces an attack within the safety limits of the individual physical and network components. In other words, we assume that individual components aren't performing actions that can damage themselves in isolation, even if they're subject to incorrect commands.

A first natural mechanism to detect anomalies is to determine whether measurements are physically meaningful locally. A more sophisticated form of control is to compare measurements and commands through physical models of the infrastructure that describe the relationship between electrical loads and the physical measurements seen during machine state changes (for example, cold or hot water changing temperature or flowing faster through the pipes, and the electrical power consumption that should be associated with these variations). This can be addressed by characterizing operational protocols using pre- and post-conditions of physical state transitions and correlating behavior from local activity captured across multiple control devices to ensure that devices adhere to the security and safety specifications.

Demonstration Scenarios

To demonstrate our approach, we implemented and evaluated a handful of initial scenarios. For each scenario, we developed and evaluated a variety of specifications of command streams in mathematical terms and implemented them in the Bro and Broccoli event languages. We installed the scripts and programs on PCs and monitored Modbus TCP and DNP3 network traffic using the Bro and Broccoli frameworks.

To examine the scenarios, we installed and configured a collection of Modbus master and slave simulation tools to analyze Modbus TCP traffic in tandem with a set of Siemens PLCs and with DNP3 simulation tools to analyze DNP3 traffic. Between the simulation tools and the PLCs, we generated Modbus TCP and DNP3 traffic to determine the requirements for detecting nonphysical data signatures and physically harmful command streams. These included elements such as the various PLC memory addresses and values in those addresses as well as the ways in which those addresses and values are commonly used among PLCs connected to cyber-physical systems. We used several physical models to create these "physical constraint" algorithms and successfully integrated them in the Bro IDS framework.

One scenario we modeled was that of a boiler, including the physics of heating and cooling when the heater is on or off and the rate of heating and cooling depending on the water level. We created Bro scripts to passively track boiler behavior and alert us to out-of-range conditions, in addition to monitoring the communication between master and slave PLCs. When Bro observed traffic that contained an entry in the Modbus TCP header with a predefined PLC memory address and a value in that address such that the temperature would be set above a specified amount, it set off an alarm when the input was recognized. We also implemented a control theoretic approach that monitors input commands and keeps track of time, internal heat, outside temperature, and water level. The IDS compares the action defined in the Modbus TCP packet (the change in the sensor values) with the allowed transition (transitions that lead to good states, derived from the control system). Any transition that doesn't lead to a good state is a potential failure or attack on sensors or actuators, and the security system raises an alert.

Another scenario involved the differential protection scheme of a power transmission line to protect the electrical power transmission line from excessive phase variance.[12] A differential protection scheme compares the phase values at two points in the line, and a difference in the phase values implies a fault in the line. We used two relays connected to circuit breakers to measure the phase difference, which signifies an internal fault where the circuit breakers isolate the transmission line from the transmission system.

Figure 1 shows this scenario's network and transmission topology. We used the Broccoli interface to support model execution as a separate parallel process, and we implemented the security system for the differential protection scheme in three major stages. In stage 1, based on the data sent from the Bro scripts, the master Broccoli script generates a well-organized network map. In stage 2, the local Bro scripts monitoring the traffic between relays analyze the packet for simple errors and protocol violation. In stage 3, a Broccoli script that has knowledge of the system's physical process understands what happens to the physical system for different commands being exchanged by the PLCs. After receiving

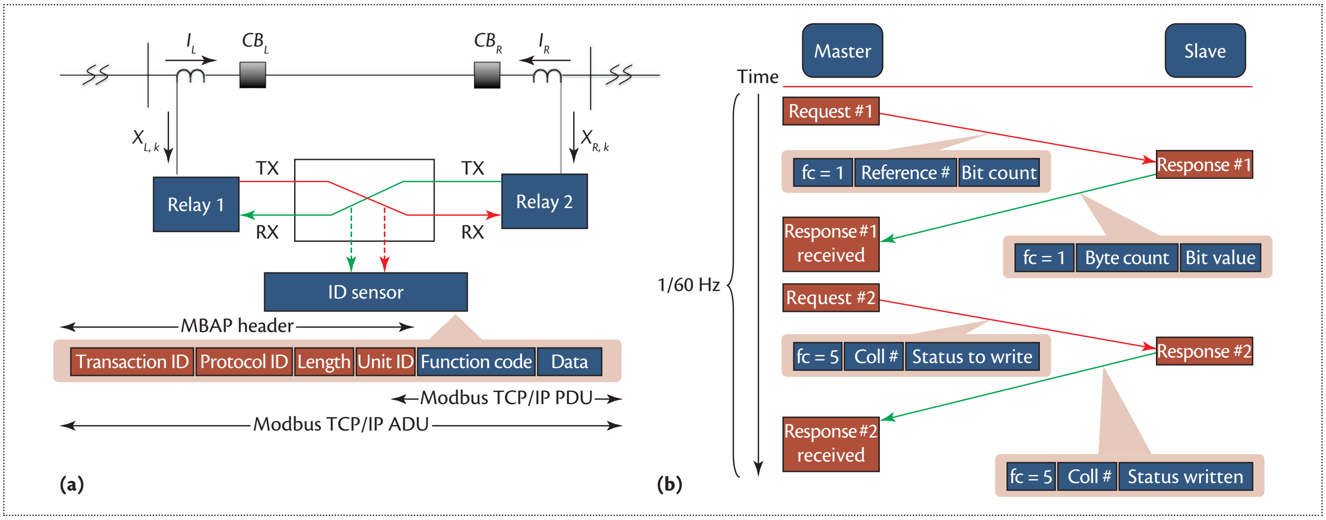

Figure 1. Scenario for differential protection. The circuit breakers CB1 and CB2 are activated if the currents don't match.

packet data from the local Bro scripts, the Broccoli script checks for vulnerabilities at the physical level by comparing phase current magnitudes.

In addition to these two scenarios, we created several other examples, including an electrical distribution fault location, isolation, and service restoration (FLISR) model (both PLC and simulation implementations). By monitoring the associated DNP3 traffic, we showed that this model allows useful real-time comparisons between relay state and line fault indicators and a simplified topological model of an actual feeder circuit.[13] By developing Bro scripts that are cognizant of the required operation of the FLISR system, we were able to evaluate a variety of attacks against the FLISR system that would cause it to fail to respond appropriately.

Using these examples, we were able to demonstrate the basic ability of the Bro-based specifications to detect command sequences that affect the overall physical operation of both the boiler and the circuit in their respective scenarios. However, there are challenges to broader applications of this system, numerous open questions, and a significant number of scenarios for which other security systems would also be needed to augment our approach. Furthermore, given the wide variety of protocols and solutions used in industrial control, a scalable approach requires an ability to learn some of the rules of the information flow in normal conditions.

Challenges and Open Research Questions

One challenge is the manual effort that's currently required to develop the specifications of the physical device limitations so that they're useful in practice. Our hope is that, over time, sufficiently large libraries of such devices will supply a critical mass available to intrusion detection, in combination with recent control theoretic approaches that have shown promise.[14] We're working to estimate the level of predictive fidelity required from these models to achieve reliable detection rates.

Our research has also demonstrated that the specifications can often be broken into "classes" of devices under control so the process can be abstracted and modularized, speeding up the creation of new specifications in given classes. Eventually, software frameworks could be constructed to support this. We're also exploring the idea of using a separate language, such as Matlab, for the physical specifications to provide a more natural means of specifying the mathematics of physical processes than the language used by Bro or C programs employing the Broccoli API. In this case, Bro would serve only as a means to parse the Modbus headers and would pass the actual physical processing to a Modbus program.

Finally, we're exploring automatic translation of vendor-supplied specifications and simulations (for example, in Simulink) to specifications, mirroring the evolution of antivirus and traditional IDSs in which malware was once largely classified by hand and eventually became mostly automated. To enable this, regulations or utility-vendor relationships could eventually require the delivery of physical device specifications along with the control device itself.

Another aspect that hasn't been sufficiently covered by present cyber-physical security research is how to guarantee that it's impossible to maliciously exploit the action space of a particular networked physical system in unpredictable ways when looking at the safety of the individual machine operations. Machines connected to the grid are tested for safety and certified in isolation, but the threats to the networked system might be larger than the sum of each possible individual threat. Any control theorist would confirm that the problem of how to divide and conquer optimal networked control has proven to be exceptionally complex. Classical examples show that policies for optimal operations are undecidable, even in simple instances.[15] Although some recent literature has a positive outlook,[16,17] modular and scalable solutions of networked control are still elusive in many cases.

The most difficult class of problems in networked control arises when it's impossible to separate control and communication timescales. In these cases, networked control problems become intrinsically very complex, if not completely intractable, due to the lack of modularity between communication and control. It's long been known from Witsenhausen's celebrated counterexample that, in these cases, the separation of estimation and controller design fails to hold even in the simplest settings.[15] Detecting control system directives that aren't of benign intent becomes increasingly difficult. The intrinsic complexity of isolating the optimal policies, by identifying the key decision variables, is independent of whether the goal of the optimal networked operations is benign. Unfortunately, this doesn't mean that the system is difficult to attack, because the real attacker has the luxury of random trial and error. This is a difficulty primarily for the designer, who can't easily decide what an attack might look like, as the system to be protected is operating in a nonideal, real-world environment and might proceed along nonideal control paths in pursuit of its operational goals.

Also of importance is the question of whether the IDS itself can be spoofed with false data or if its alarms can be spoofed to the operator. Either scenario would result in a common mode failure, in that the IDS could be used to attack the system. The first scenario can be partially addressed using a control theoretic approach in which expected behavior is predicted and compared against reality. Unfortunately, if the device is manipulated to transmit sensor data that appears normal but the physical device is behaving abnormally, our technique likely can't catch this. The notion that IDS alerts can be spoofed and sent to operators is also a risk for any IDS and can be difficult to manage. One strategy is to position the IDS in such a way that it observes traffic (for example, via a switch's SPAN port) but operates on a more isolated communication channel so that it's not directly addressable on the same network as the control devices. Although this doesn't prevent attacks, it might be a mitigation strategy.

Bro currently has limited capabilities to act as a prevention system in the ways we've described in this article because it's a passive monitor and isn't used as a flow-through system. When Bro is used as an actual IPS, it isn't used to block packets in real time, but it can be configured in near real time to update routers with new access control lists or firewalls with new rules. However, this wouldn't prevent a command from taking place in real time. If it were to act as a flow-through system in this way, additional timing challenges relative to SCADA systems would present key issues to work through.

Although there are significant challenges and open research questions regarding using this technique in practice, we've already taken certain steps to address many key questions and believe that our technique has demonstrated promise.

By leveraging an understanding of the physical limitations of cyber-physical systems under control as well as the protocols used to monitor and send commands to such devices, specification-based intrusion detection can be used to monitor a cyber-physical system to verify that it operates according to the specifications of the networked physical system being controlled. This emphasis on melding security and safety bridges the gap in existing network security techniques to mitigate damage to cyber-physical devices. Our method, using the Bro IDS, demonstrates that combining this concept into a single, reliable IDS framework results in a highly capable, vertically integrated, and context-sensitive cybersecurity tool for use with SCADA control systems. ■

Acknowledgments

This research was supported in part by the Director, Office of Computational and Technology Research, Division of Mathematical, Information, and Computational Sciences of the US Department of Energy, under contract DE-AC02-05CH11231. It was also supported in part by the National Science Foundation under grant CCF-1018871. Any opinions, findings, conclusions, or recommendations expressed in this article are those of the authors and do not necessarily reflect those of any of the employers or sponsors of this work.

References

1. X. Li et al., "Networked Loads in the Distribution Grid," *Proc. APSIPA Annual Conference*, 2012.
2. N. Falliere, L.O. Murchu, and E. Chien, "W32.Stuxnet Dossier (v.1.4)," Symantec, 2011; www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf.
3. J. Slay and M. Miller, "Lessons Learned from the Maroochy Water Breach," *Critical Infrastructure Protection*, vol. 253, 2008, pp. 73–82.
4. U.S. Department of Energy Office of Health, Safety and Security, *Russian Hydroelectric Plant Accident: Lessons to be Learned*, 4 August 2011.
5. I. Ghansah, "Smart Grid Cyber Security Potential Threats,

Vulnerabilities and Risks," California Energy Commission, PIER Energy-Related Environmental Research Program, CEC-500-2012-047, 2009.

6. R. Berthier, W.H. Sanders, and H. Khurana, "Intrusion Detection for Advanced Metering Infrastructures: Requirements and Architectural Directions," *1st IEEE Intl. Conf. Smart Grid Comm.*, 2010, pp. 350–355.
7. Á. Cárdenas et al., "Attacks against Process Control Systems: Risk Assessment, Detection, and Response," *Proc. 6th ACM Symp. Information, Computer and Communications Security*, 2011, pp. 355–366.
8. Y. Mo et al., "Cyber-Physical Security of a Smart Grid Infrastructure," *Proc. IEEE*, vol. 100, no. 1, 2012, pp. 195–209.
9. C. Ko, M. Ruschitzka, and K. Levitt, "Execution Monitoring of Security-Critical Programs in Distributed Systems: A Specification-Based Approach," *Proc. IEEE Symp. Security and Privacy*, 1997, pp. 175–187.
10. R.S. Boyer, M.W. Green, and J.S. Moore, "The Use of a Formal Simulator to Verify a Simple Real Time Control Program," *Beauty Is Our Business: A Birthday Salute to Edsger W. Dijkstra (Monographs in Comptuer Science)*, 1990, pp. 54–66.
11. V. Paxson, "Bro: A System for Detecting Network Intruders in Real-Time," *Proc. 7th Usenix Security Symp.*, 1998.
12. G. Koutsandria et al., "A Hybrid Network IDS for Protective Digital Relays in the Power Transmission Grid," *Proc. 5th IEEE Int. Conf. Smart Grid Communications*, Nov. 2014.
13. M. Parvania et al., "Hybrid Control Network Intrusion Detection Systems for Automated Power Distribution Systems," *Proc. 1st Int. Workshop Trustworthiness of Smart Grids*, 2014, pp.774–779.
14. R. Chow et al., "Enhancing Cyber-Physical Security through Data Patterns," *Workshop Foundations of Dependable and Secure Cyber-Physical Systems*, 2011.
15. H.S. Witsenhausen, "A Counterexample in Stochastic Optimum Control," *SIAM J. Control*, vol. 6, no. 1, 1968, pp. 131–147.
16. L. Schenato et al., "Foundations of Control and Estimation over Lossy Networks," *Proc. IEEE*, vol. 95, no. 1, 2007, pp. 163–187.
17. S. Tatikonda and S. Mitter, "Control under Communication Constraints," *IEEE Trans. Automatic Control*, vol. 49, no. 7, 2004, pp. 1056–1068.

Chuck McParland recently retired from Lawrence Berkeley National Laboratory, where he had been a staff computer scientist since 1979, working on projects ranging from data acquisition from satellites to neutrino sensors buried under the Antarctic ice. McParland received bachelor's degrees in physics and philosophy from the University of California, Berkeley. Contact him at cpmcparland@lbl.gov.

Sean Peisert is a staff scientist at Lawrence Berkeley National Laboratory and an assistant adjunct professor at the University of California, Davis. His research in computer security includes intrusion detection and vulnerability analysis. Peisert received a PhD in computer science from the University of California, San Diego. He's a senior member of IEEE and the ACM. Contact him at sppeisert@lbl.gov.

Anna Scaglione is a professor of electrical and computer engineering at the University of California, Davis. Her expertise is in the broad area of signal processing for communication systems and networks. Scaglione received a PhD in electrical engineering from the University of Rome La Sapienza. She's the recipient the IEEE Donald G. Fink Award and is a Fellow of IEEE. Contact her at ascaglione@ucdavis.edu.