UC Davis

UC Davis Previously Published Works

Title

Designed-in Security for Cyber-Physical Systems

Permalink

https://escholarship.org/uc/item/3d8895mn

Journal

IEEE Security & Privacy, 12(5)

ISSN

1540-7993

Authors

Peisert, Sean
Margulies, Jonathan
Nicol, David M
et al.

Publication Date

2014

DOI

10.1109/msp.2014.90

Peer reviewed

eScholarship.org Powered by the California Digital Library
University of California

Designed-in Security for Cyber-Physical Systems

Sean Peisert | Lawrence Berkeley National Laboratory and University of California, Davis
Jonathan Margulies | Qmulos
David M. Nicol | University of Illinois
Himanshu Khurana | Honeywell
Chris Sawall | Ameren

The November/December 2014 issue of *IEEE Security & Privacy* will focus on control system security in the energy industry. As a preview, guest editors Sean Peisert and Jonathan Margulies hosted a roundtable discussion featuring three experts from the energy sector—David M. Nicol, Himanshu Khurana, and Chris Sawall—who offer different perspectives on the meaning and challenges of "designed-in security": one from academia, one from a cyber-physical system (CPS) provider, and one from an end asset owner and user.

Like the parable of the blind men and the elephant, we get three fairly distinct viewpoints. The academic highlights foundational issues and talks about emerging technology that can help us design and implement secure software in CPSs. The provider's view includes components of the academic view but emphasizes the secure system development process and the standards that the system must satisfy. The user issues a call to action and offers ideas that will ensure progress.

An Academic's Perspective: David M. Nicol

There are at least two types of security vulnerabilities: flawed or insufficient software behavior specifications that allow compromise—for example, authentication that is lacking or that can be spoofed—and incorrect implementation of specifications—buffer overflows being a classic example and Heartbleed being a notable recent one.

The sad truth is that the state of the art in specification development and implementation doesn't deliver secure software. So the question is, what does designed-in security mean in each case—for CPSs in particular—and what are the challenges?

For software implementation, designed-in security includes programming language features that enhance security. Many security-enhancing programming language ideas have been proposed, but none have had traction in CPSs. Type-checking of data objects passed through interfaces is an example; at compile time and potentially at runtime, the runtime system analyzes which objects pass between software modules or routines and ensures that what's passed looks like what's expected. Types can have attributes, which let us specify interfaces whether encryption or authentication is required.

However, this kind of dynamic introspection takes time. A CPS's software component usually runs in a real-time control loop, which means it must be fast enough to keep up with a schedule. In addition, the software must have a predictable execution time, and features such as automatic garbage collection impede that. Yet, dynamic memory management errors are a common source of vulnerabilities.

My hope is that we can discover a sweet spot in the spectrum of programming language features that gives real-time system designers sufficient speed, predictable behavior, and features that enhance security. I would also hope for widespread adoption of such a language by the energy industry. Frankly, I think the latter problem is more difficult owing to the many understandable but very real business impediments to change, such as the immaturity of such technology and a lack of vendor support.

1540-7993/14/$31.00 © 2014 IEEE Copublished by the IEEE Computer and Reliability Societies

Our Roundtable Participants

David M. Nicol

Himanshu Khurana

Chris Sawall

When I think of designed-in security for system specification, I think of various types of system models and the interaction of components and security policies, for example, formal languages that model a protocol and prove certain properties about it. The academic landscape has many knolls of activity outside CPS, and in some (usually limited) cases, impressive results are possible, such as formal verification of programs written in a particular subset of the C language. However, adding a physical component introduces new complexities and requirements for such formal analyses, because the environment in which the software runs can affect system behavior, which itself can be an attack vector. Manipulated, corrupted, or spoofed sensor reading values could push the software system into a state that it's designed to enter—perhaps unintentionally—and have deleterious consequences.

More research is necessary in areas such as increasing confidence in sensor readings by checking consistency with other sensors and information sources as well as validating control system commands by precomputing their impact using a physical system model. Some of these goals are now reachable at a rudimentary level in an industrial context—for example, Khurana and I worked on a proposal to the Department of Energy to perform consistency checking on electrical distribution system commands. However, I don't see sufficient business drivers for the energy industry to develop this kind of research. The government will have to lead.

A Provider's Perspective: Himanshu Khurana

CPSs have computational and physical components where sensing, data, analytics, and control come together to add value across a range of industries, including the smart grid, intelligent buildings, healthcare monitoring, autonomous vehicles, and smart manufacturing. These systems, built from various engineering components, must realize application-driven functionality and satisfy a range of nonfunctional properties including safety, reliability, and security. Security is a key risk and concern but suffers from a lack of precise objectives, solutions, and measures.

Today, the industry's key issues are defining designed-in security, designing and developing accordingly, dealing with the absorbed risks, and incorporating improved solutions as they emerge. If we do these well, we can establish good synergy between industry deployment and research improvements. We observed this cycle in the IT industry; however, things might play out differently in the CPS industry owing to its product life cycles—for instance, there's still a gap between consumer-driven two-year smartphone upgrades and that sensing and control box in the ceiling of every floor of every commercial building, which might not be replaced for another decade. Some people call this the legacy problem, but it's more than that—though perhaps this is a topic for another conversation.

Let's break down the problem of designed-in security for CPS into a few bite-sized chunks. From a systems perspective, designed-in security involves, at a minimum, requirements, solution engineering and composition, and verification and maturity (or improvement).

Security requirements come from various sources such as regulation, government-sponsored frameworks, industry consortiums, standards, asset owner specifications, and so on. These requirements are a good step forward and facilitate increasingly secure solutions. There's a lot of overlap across these requirements frameworks, and as we develop them further, we should explore the realization of a common framework. Furthermore, to address boundary issues, the

frameworks should clarify roles and responsibilities across the solution chain, from asset owner to component provider.

Realizing solutions typically involves combining many components sourced from many providers and then deploying, configuring, and managing them correctly. Components include sensors, actuators, computational hardware, firmware, operating systems, applications, and networking subsystems. We worry about the security of smartphone systems (and rightfully so) that are developed by a few key vendors and manufactured at a few key sites. Compare that to the plethora of CPSs that might be developed by hundreds of providers and manufactured in as many sites. Without doubt, a subset of CPS components won't satisfy desired security properties, so part of the challenge is designing with a little bit of insecurity. We can manage this with careful architectural considerations and by employing roots of trust for key decisions and control functions.

Secure composition methods play an important role; Nicol outlined some nice development methods and tools currently under exploration. In IT systems, we often address component and configuration weaknesses with continuous security monitoring and incident response. Adoption of equivalent CPS solutions is increasing, and this is a good direction for further efforts.

I often think about verification and maturity as a continuum. Today, we verify CPS security using threat modeling, code reviews, and various white/gray/black-box testing tools, giving us a sense of requirements compliance, significant mitigation of known attack vectors, and somewhat limited mitigation of unknown attack vectors. We can enhance this approach by integrating maturity goals so we always know how much we want to improve in the next version, be it a product or network. To that end, I support recent maturity framework development efforts in CPS security. As researchers develop more quantifiable metrics and practical assessment tools, we'll see improvement in verification as well as our confidence in the security of the overall solution.

A User's Perspective: Chris Sawall

Although I agree with what my partners said, I feel we could be more simplistic. Designed-in security means that all products, solutions, technology, and so forth, are designed with security concepts in mind. We should employ a security development life cycle. Many companies state that quality is the number-one goal, yet they don't always consider basic security requirements. Isn't cybersecurity a core requirement for a quality product? Without it, as both Nicol and Khurana stated, there are flaws—potential for system compromise or failure.

Designed-in security should extend beyond the individual device or application to the entire system (or group of devices or networks). Many control systems are implemented as a complete turnkey system; designed-in security must apply to this entire system. Although control systems' core functionality is to run a plant or other critical infrastructure or manufacturing facility, they must be resilient, safe, secure, and reliable. A built-in layered security model would help prevent any one system failure or compromise from having a cataclysmic effect on the entire control system.

However, cybersecurity comes at a cost. Generally speaking, vendors put into their products only what their customers want and what they know will sell; they want a return on investment. The industry is very proactive in protecting assets. In fact, many companies in the energy sector have asked that security be integrated into their vendor solutions, and many vendors have heard the call to arms. It took a while to get movement, but the movement has begun and is a great first step.

But what next? Are cybersecurity controls in one solution as good as another? Must companies patch critical security vulnerabilities or face a voided warranty? Some industry professionals have discussed an "underwriter's laboratory"–like evaluation and "seal of approval" for vendor solutions and technology. A centralized entity that reviews technology to verify key security controls would be an excellent step forward, providing the industry with normalized validation that ensures a robust solution with security designed into it. In addition, this centralized entity could help validate that patches and upgrades don't compromise system stability.

Creating a group like this won't be easy. And there are several concerns with this thinking, such as the fact that it's only a point-in-time evaluation. A worst-case scenario is creating a false sense of security whereby consumers believe they're protected, with a validated and approved security architecture and system, yet fail to understand that attackers will always work to be one step ahead and will eventually find a flaw to compromise. Perhaps a middle ground exists.

Energy is one of the strongest critical infrastructure sectors, with significant peer collaboration as well as several committees, working groups, and research projects.

Only by working with vendors and other partners on a common goal will the industry solve this problem. The industry can't do it alone, and vendors can't work in silos assuming they know what's needed or what will work.

Great work has already been done and a vision set. In 2011, the Energy Sector Control Systems Working Group published the "Roadmap to Achieve Energy Delivery System Cybersecurity" (http://energy.gov/sites/prod/files/Energy Delivery Systems Cybersecurity Roadmap_finalweb.pdf), which offers the following vision: "By 2020, resilient energy delivery systems are designed, installed, operated, and maintained to survive a cyber incident while sustaining critical functions." Without designed-in security or industry collaboration, we have no chance of achieving this vision.

We thank the panelists for taking the time to give their perspectives on the meaning and challenges of designed-in security as it pertains to cyber-physical systems in the energy sector. The insights from everyone, as well as the distinctions, have made for a very enlightening discussion. ■

Sean Peisert is jointly appointed as a staff scientist at Lawrence Berkeley National Laboratory and as an assistant adjunct professor at the University of California, Davis. His research interests cover a broad cross-section of computer and network security. Peisert received a PhD in computer science from the University of California, San Diego. Contact him at sppeisert@lbl.gov.

Jonathan Margulies is director of analytics at Qmulos. Contact him at margulies@gmail.com.

David M. Nicol is the Franklin W. Woeltege Professor of Electrical and Computer Engineering at the University of Illinois, and director of the Information Trust Institute, where he manages and conducts research related to electric power grid security. Nicol received a PhD in computer science from the University of Virginia. He's a Fellow of IEEE and the ACM. Contact him at dmnicol@illinois.edu.

Himanshu Khurana is director of engineering at Honeywell Building Solutions. He focuses on technology strategy and product innovation for integrated building management systems with an emphasis on security, energy, operations, and safety. Khurana received a PhD in electrical and computer engineering from University of Maryland, College Park. Contact him at himanshu.khurana@honeywell.com.

Chris Sawall did this work while he was the director of cybersecurity at Ameren Services Company. His research interests include cybersecurity across all business segments and for regulatory cyber-related compliance. Sawall received a BS in telecommunications management from DeVry University and a Certified Information Systems Security Professional certification from ISC2. Contact him at sawall@gmail.com.

Selected CS articles and columns are also available for free at http://ComputingNow.computer.org.