IEEE CYBER SECURITY

Building Code for the Internet of Things

Ulf Lindqvist and Michael Locasto
SRI International

IEEE IEEE computer society

READ ON!

Public Access Encouraged

Staff

Brian Kirk, Manager, New Initiative Development
Jennie Zhu-Mai, Designer

TABLE OF CONTENTS

Executive Summary

The Internet of Things (IoT) is already part of our daily lives, and will become even more so in the near future. The many characteristics that make the IoT different from the traditional networked computing, such as the close interaction with the physical world, also require us to pay particular attention to how to make such systems safe and secure. This document is part of a series that has previously addressed how to build more secure medical devices, connected vehicles, and electric power systems. In this document, we focus on the challenges associated with composing systems, rather than building individual programs or devices. We use the concept of *smart cities* to illustrate how design for safety, security, and privacy must consider emergent properties, and how a system or technology designed for this domain must account for how it might be integrated, reused, or composed with other technologies and systems.

The IEEE Cybersecurity Initiative organized a workshop where a group of invited experts focused on the particular challenges to safety, security, and privacy that emerge from considering the interactions between heterogeneous IoT systems that are integrated to implement the smart cities concept. This document represents the outcome of that workshop and is intended to provide guidance to system vendors, integrators, and municipal officials and citizens who are stakeholders in smart cities. Here, we describe fundamental challenges related to system composition and unpredictable interactions, followed by a categorized set of topics and questions for integrators and policymakers to consider in the design and deployment of IoT technology in municipal infrastructures. The questions put forward in this document are important to ask prior to the development and deployment of IoT systems of systems that could greatly affect people's lives.

BACK TO TABLE OF CONTENTS

Introduction

The IEEE Cybersecurity Initiative has organized a series of invitational workshops to establish an initial consensus among industry and academic participants on the appropriate components of a "building code" intended to significantly reduce vulnerability to cyberattacks.[1] The Internet of Things (IoT) is a broad concept that comprises all kinds of "things" equipped with computing and communication capabilities, including the types of systems that have been the focus of previous workshops in this series: medical devices,[2] connected vehicles,[3] and electric power systems.[4]

The objective of this workshop was to leverage the knowledge and structure gathered from the experience of these past workshops to provide insight into methods for writing building code requirements that would apply to systems of systems. The workshop focused on the particular challenges that emerge from considering not only a larger scale, but the particular methods and practices that govern interactions between the heterogeneous systems that comprise the larger IoT. The concept of *smart cities* was used as a framework and guiding use case for the discussion.[5,6] Visions for the use of IoT to implement smart cities include applications such as monitoring the structural health of buildings; waste management and recycling; air quality and other environmental monitoring; noise monitoring; traffic monitoring and control; energy consumption; hazard detection and alerting; parking; lighting; and building automation.[7] Interoperability among heterogeneous systems is a general technical challenge in implementing such IoT systems of systems. From a cybersecurity perspective, developers need to know how to create secure design and implementation that supports heterogeneity and emergent interactions.

BACK TO TABLE OF CONTENTS

IoT and Smart Cities Context

IoT isn't the only technology that's key to implementing smart cities, but it's certainly dominant in terms of rapid development and deployment, and also in terms of challenges with respect to integration and security. The sensing, data collection, and other decentralized interactions with the physical world that IoT can provide, particularly interactions with the people living in and visiting the city, are essential to implementing the smart cities vision. It's therefore very important to understand how to secure various aspects of IoT against malicious attacks that could threaten the confidentiality of data and integrity and availability of data and services. While aspects of how to build secure IoT systems are being studied, including in the previous workshops and reports in this series, the big challenge of how to securely compose large-scale systems consisting of heterogeneous IoT systems has until now largely been ignored. This challenge is difficult because of the problem space's scale and complexity, and further complicated by the multitude of owners and competing stakeholders inherent in any composition problem. Yet, latent and unaddressed liability concerns in this domain suggest that a consideration of these issues is necessary ahead of a serious, widespread malfunction or successful cyberattack with physical-world impact.[8]

Smart cities were chosen as an application domain because they're a good example of "systems of systems"—a collection of IoT systems that in themselves can be complex, but also need to compose and interact well as a larger collective system to provide not only convenience and efficiency but also safety, security, and privacy for the people. Because today's smart city systems are often pilot deployments that are intended

BACK TO TABLE OF CONTENTS

by vendors and visionary city officials to demonstrate capabilities (and help vendors in their marketing to other municipalities), they're typically not designed to deal well with disruptions, such as power or Internet outages, or intentional cyberattacks. The many aspects of life in a metropolitan area are connected and interdependent, and therefore smart city systems similarly will be interconnected and interdependent. For example, traffic monitoring systems will provide information about the timing and intensity of commuter traffic and congestion, which could inform building energy management systems about the expected occupancy load and also public transit systems about expected delays. Realizing the potential benefits of these kinds of interactions demands that their safety and security properties be well understood; there's no commonly adopted discipline or set of requirements governing the future relationships between complex systems and devices that arise from unintended or serendipitous composition.

A major challenge for smart cities is that subsystems designed and implemented in isolation will lack support for security mechanisms required by the larger interconnected system. In this work, we seek to identify what those security mechanisms are and how their properties and requirements differ from traditional security mechanisms for less complex systems.

BACK TO TABLE OF CONTENTS

Purpose

The purpose of this document is to present guidelines to system vendors and integrators, and also to municipal officials (customers), on methods for dealing with security and privacy considerations that arise from connecting and federating IoT technology across public and private municipal infrastructures to implement smart city concepts. This document represents the outcomes of an IEEE Cybersecurity Initiative invitational workshop held 18–20 May 2017 at SRI International in Menlo Park, California.

This document complements the published reports on secure coding guidelines that resulted from previous workshops in this series on medical devices,[2] connected vehicles,[3] and electric power systems.[4] Rather than exhaustively covering all aspects of building secure IoT systems, this document focuses on the issues specifically related to composition and safe emergent interaction, as viewed in the smart cities context.

Workshop participants were encouraged to view this problem space in light of the five phases of the system development life cycle (SDLC) process, which is the overall process of developing, implementing, and retiring information systems in the phases of initiation, acquisition/development, implementation/assessment, operations/maintenance, and sunset/disposal.[9]

This document isn't really a building code for building code, in the sense that it doesn't describe how to build a single software program (let alone a collection of software) that's secure from a particular class of attack. This document's content doesn't describe how to improve code quality. As a building code for composition, it's instead a set of topics and questions for integrators and policymakers to consider in designing and deploying security mechanisms specialized to the task of enforcing security and providing resiliency across things with non-coordinated designs.

BACK TO TABLE OF CONTENTS

Challenges

First, we note that interaction between IoT devices and systems—including unpredictable, emergent interactions among otherwise unassociated devices and infrastructures—is in fact a desirable outcome in this setting. IoT's unique value proposition is based on these types of interactions being possible: if IoT is anything more than a few Internet-connected home appliances, or in some way categorically different than a handful of scattered "smart" devices, it's a vision for how ubiquitous computing and networking enable intelligent processing of a variety of data streams and seamlessly actuate significant portions of our physical world. In short, we expect this next iteration of the Internet to be able to find and solve new and unique problems for us with relatively little oversight or traditional "programming." Therefore, our primary challenge is to determine how we can ensure that such interactions take place in a fashion that meets security, safety, and privacy requirements.

For devices and their owners, the main challenge is "trustworthiness estimation" at scale. This challenge can be stated as the following practical concern:

> *Upon entering an environment, how does a device ascertain that communications and physical interactions are safe in both benign and malicious conditions?*

System owners and designers need a collection of procedures that establish a foundation for and the presence of these safety properties relatively quickly.

Further complicating matters, we posit that the purpose for which individual systems are designed (including the design of any security mechanisms they incorporate) is different from

BACK TO TABLE OF CONTENTS

how they later could be used in conjunction with legacy, current, and future systems. Individualized designs tend to not anticipate collective weaknesses. As a potential counterbalance for this difficulty, however, there's an opportunity for different subsystems to compensate for others during disruptions. We can therefore ask:

> *What elements and what security and privacy mechanisms do we need in the smart cities setting that anticipate and compensate for emergent security and privacy weaknesses in the entire system of systems?*

Another challenge is to determine what should be required of device creators to certify their systems for proper operation in a smart cities setting under benign and malicious conditions:

> *What are the social, legal, and ethical responsibilities for device creators and system integrators with respect to anticipatory design?*

In this document, we lay no onus but that of professional responsibility and convention.

BACK TO TABLE OF CONTENTS

Elements of the Code, by Category

The code presented here is categorized differently than in the previously published building codes in the IEEE series, because in this document we focus on the composition of systems, rather than building individual programs or devices. For every category or code element presented here, we provide a technical description illustrated with examples, and also a list of questions that municipal officials and other smart city system customers and stakeholders should ask the system providers with respect to that particular topic.

Responsive Analytics

In response to a detected extraordinary or emergency event, a component could make unanticipated use of physical interconnects between smart city subsystems and use service and actuator discovery mechanisms to enable data stream collection from multiple systems and perform analytics that weren't originally planned.

For example, an automatic gunshot detection and location system could, upon detection of gunshots, request traffic cameras and other nearby sensors to start recording and target relevant areas so that subsequent activity could be recorded and fleeing criminals could be identified and tracked, and traffic lights and other controls could be used to make it easier for law enforcement to catch up with the perpetrators.

Questions for system providers

- How do individual systems securely advertise their physical and computational capabilities?

BACK TO TABLE OF CONTENTS

- What authorization mechanisms are in place to control system responses to requests from other systems?
- How can denial-of-service attacks exploiting the request mechanism be mitigated? Can a cost be associated with making a request (such as cryptocurrency or proof-of-work) without impacting the scalability of legitimate high-volume traffic? Could an ongoing denial-of-service attack be detected?
- How can systems that issue help requests be protected from compromise by untrustworthy helper systems?

Safe Autonomy

We typically want systems to have some degree of autonomy, but when system actions significantly impact the physical world, we would like to be able to limit and monitor that autonomy. The security design pattern to be applied here is continuous complete mediation (a la Jerome Saltzer and Michael Schroeder[10]), but the nature of the security monitor is different (indeed, it might have an impossible task); to safely limit autonomy, a monitor must make some calculation about the potential physical side effects of the program decision. The security decision is transformed from checking a credential against an access list to performing what might be a computationally unbounded procedure. Because this introduces the possibility of undecidable computation into the security monitor, the challenge here must be to state the security decision's limits, and how this attests to the control provided so that user expectations are appropriately set. Every autonomous action must be referred to an independent competent authority, which must receive and process an action credential that attests to the action's propriety to be performed and its binding to the entity proposing to perform it.

Questions for system providers

- To what extent does your system have autonomous functions?
- To what extent can autonomy safely be limited?
- How can decisions made autonomously be reviewed and approved or blocked before they take effect?

Namespace Control Translation

In communication between systems, there needs to be agreement among all communicating parties (systems) on symbols and their meaning. For example, if a particular label (such as `event_start_time`) has a different format (syntax) and/or meaning (semantics) in two different systems, when those systems exchange information using that label, problems will arise because of the assumed shared understanding

BACK TO TABLE OF CONTENTS

but actual differences in interpretation. Another example would be two systems using the same namespace to assign the device ID, which would work fine within a system but wouldn't work to address devices across system boundaries. Such differences or overlap in namespaces could also be exploited by attackers.

One approach to address this issue without the need for all systems to agree in advance on a global namespace, would be to involve a high-assurance translator function that's authoritative and understands syntax and semantics on both sides of a data interconnect and can provide translation services as needed. This approach is comparable to network address translation (NAT), which is a technique commonly used on the Internet to "hide" subnetworks with private IP addresses that aren't globally unique behind a single globally reachable IP address. An alternative approach would be to design systems that could directly negotiate with each other using a prescribed common language, but that could require more resources than some IoT systems have.

Questions for system providers

- How are the external communication interfaces and their namespaces specified and documented?
- How would a translator function be supported?
- How would direct negotiation with other systems be supported?

Data and Action Provenance

Data and actions that affect people's lives in smart cities should carry provenance data showing their origin and history, to enable detection of errors and unauthorized manipulation, to provide accountability, and to enable recordkeeping and auditing. IoT devices that compose smart cities subsystems are likely to come from different manufacturers and include a large variety of software from many vendors and application suppliers. Nevertheless, to achieve interoperability, these devices need to exchange data. As data crosses these design boundaries, it's likely stripped of meta information or other provenance data. This is because copying or replicating bytes of information is a basic, easily accomplished operation in most systems, and in the name of abstraction, most implementers might simply locate the information they think they want or is sufficient to their needs, copy it into their own system, and begin operating on it.

An example of data carrying provenance information includes temperature measurements that are signed, associated with attested GPS coordinates and a timestamp. Any computations or actions that change or add information are documented in the same way.

BACK TO TABLE OF CONTENTS

Questions for system providers

- Do you provide a format or data dictionary service for storing provenance data?
- Is this data service available and described for third parties?
- How is provenance data linked with raw data transmitted by the system?
- Does your incoming input system boundary require information suppliers to provide some form of provenance data?
- Does your system include an integrity monitoring and provenance checking functionality that examines any supplied provenance or metadata?

Flexible Isolation Boundaries

Most systems designed with security requirements mandate or have implicit security boundaries to enable the separation between "good" and "bad" that's at the heart of most security requirements. Security boundaries can be expressed in a variety of ways, but often they're defined by a collection of monitoring functions that, after making an access control decision, permit the invocation of a certain system functionality or reading or writing system data. The assumption is that, once identified, sensitive information and functions are sequestered within this perimeter, and all legitimate access must pass through an entry function (a gate) that forms the boundary. This notion is behind most of the foundational work on protection levels from early computer and operating system security. In that problem setting, memory and data areas had an underlying well-defined notion of *space*: a logical address range with a starting address and a known offset or limit. Over the decades, this concept has been stretched so that it's now common to refer to entire virtual machines as being within some protection boundary; this protection might be afforded to only one aspect of the system (for example, the network) and for only one type of control (such as the source IP addresses or incoming ports).

Even in the case where isolation boundaries are well-defined, complete, and sufficient to protect a system or component against compromise, interacting IoT systems might require well-defined ways of adjusting this isolation to access parts of another system (for example, in the case of a smart cities subsystem compensating for another during a natural disaster). The decision process governing this adjustment of the isolation boundary needs to be able to gauge the context of the situation and the trustworthiness of the entities being considered for inclusion inside the boundary.

Questions for system providers

- How is the security boundary of your system defined? What processes does your

BACK TO TABLE OF CONTENTS

system contain to enumerate or outline that boundary, either statically or dynamically?
- How is the security of your system dependent on a strictly defined boundary?
- What mechanisms are available to adjust the system boundary?

Minimizing Functionality and Communication Expressiveness

Complexity tends to lead to poor security. The more unnecessary functionality and expressiveness that exists in a system or protocol, the more opportunities there are for an attacker to exploit one of the "bonus" features. Market economics may tempt vendors to bundle many functions together in a device or system, even though particular deployments only use a subset of those functions. First, the temptation to bloat protocols with functionality, fields, and expressiveness is in itself a problem actively being mitigated—for example, with Language Theoretic Security (LangSec),[11] and what the previous Building Code for Building Code (BC2) workshops have already noted relative to this point. Second, this problem is made even worse because the nature of the domain means that other parties will attempt to interoperate with existing, partially compliant, partially observable, or reverse-engineered descriptions of this protocol, format, or interface to achieve some interaction or other goal. Too much expressiveness then risks encouraging poorly built, unprincipled combinations of data flows.

Questions for system providers

- How are you limiting expressiveness in external communication for your system?
- How are you limiting system functionality to "need to have" rather than "nice to have"?
- What are you doing to remove or disable unneeded functions provided by the components in your system?

Predicting the Physical Impact of Virtual Connection

We note that a major difference between traditional IT and the IoT is that IoT devices are tightly connected to the physical world, often in the form of sensors but also as actuators. A virtual connection between smart devices is a channel over networking communications technology that might make use of common Layer 1 and Layer 2 media and Layer 3 networking protocols. The essence of these connections (in many cases, a "traditional" network connection that permits the devices to send and receive data to service endpoints) is that they eventually invoke or induce a sensor or actuator. In this domain, reads and writes of network messages no longer merely affect the

BACK TO TABLE OF CONTENTS

memory state of internal devices (hard disks, for example) of the receiving computer; rather, these messages go on to measure or affect the physical world. In some cases, these connections might be anticipated, intended, and well-controlled. In many others, such connections might be unpredictable, unplanned, and hard to constrain. They might occur with the incidental aid of other computers or network devices, or indirectly "hop" through another physical channel (such as a property of the environment such as temperature, pressure, sound, or light).

Some examples include IP over 802.11, IP over Ethernet, the Vendor protocol over BluetoothLE, and the ANT+ protocol over Bluetooth.

Questions for system providers

- Do you have high-fidelity models of your physical device that go beyond a simple discrete simulation of the device's inputs and outputs? Have you modeled the physical properties of the device components?
- Can the device be made to fail by sending it sequences of "legitimate" commands or values, but at varying rates, or by rapidly reversing commands, or similar unplanned sequences? (For example, could it be made to fail by flipping a power switch on and off tens or hundreds of times per second?)
- Have you described an API for a "library" of physical actuators and sensors that your system or device makes available, either directly or indirectly? What security monitors, if any, are interposed on this functionality?
- What's the expected set of devices that can access this (perhaps undocumented) API?
- What are the fail-safe considerations and manual override mechanisms for high-impact physical actuators? Can an actuator be "unplugged" physically (or disabled virtually), and if so how, and what would the impact be?

Locus of Security Responsibility and Control

When the security of a single system is under consideration, then it's easy to imagine that a portion of the system is responsible for limiting access and actions.[10] Thus, a single computer has a reference monitor, and a centralized security system with many such systems can hold the definition of a policy. In an IoT setting, it's possible that some sensors and some actuators won't be owned by the same organization. While many actions might be completely safe to perform (for example, turning on lights in the daytime to troubleshoot an infrastructure's electrical problem), other, similar actions might be unsafe to perform (such as turning off lights on a block at night time).

BACK TO TABLE OF CONTENTS

Questions for system providers

- To what extent can some control be shared with another entity?
- What are the fail-safe aspects of your design?
- If something goes wrong, who's responsible for the real-world effects?

Managing Obsolescence and Sunsetting

IoT and smart cities systems can have long planned lifetimes (10 years or more), but we also know that infrastructure systems tend to remain in use much longer than originally planned. Problems arise when new systems need to interact closely with aging technology that can't be updated for various reasons. It might not be feasible or even safe to make newer systems fully backwards-compatible with all older systems, because the older protocols could have vulnerabilities that are widely known. The company that developed or supported the old system may no longer be supporting it or even be in business. Distributed, autonomous IoT devices that are deployed in large numbers and embedded into roads, sidewalks, and buildings can be costly and difficult to disable and replace. The future could be polluted with unsupported, vulnerable devices that still operate but with limited security and reliability.[12] Plans and considerations for deploying a new system should include plans for safe sunsetting and possible replacement when the system has reached its anticipated useful lifetime.

Questions for system providers

- What's the planned useful lifetime for the system?
- How will security updates be supported if the system outlives the provider? Will development documents, source code, and the rights to make modifications be made available to the customer after the system provider stops supporting the system?
- How can the system be safely, securely, and gracefully decommissioned once it's determined to have reached its useful lifetime?
- How can collected data be transferred to a replacement system in a fashion that meets security and privacy requirements?

BACK TO TABLE OF CONTENTS

Conclusion

While this document is a source of questions rather than answers, it's important that these questions are asked before IoT systems of systems are developed and deployed. System vendors, integrators, and municipal officials are encouraged to use this document and discuss the security considerations that arise from connecting and federating IoT technology across public and private municipal infrastructures as smart cities concepts are being implemented in cities around the world.

Acknowledgments

We thank all of the workshop participants for their contributions before, during, and after the meeting. (See the related sidebar for participants' names and affiliations.) The expertise, commitment, creativity, and overall positive interaction displayed by the workshop participants made the event highly productive and enjoyable. While there are certainly many challenges in security and privacy for IoT and smart cities, this workshop gave us some hope that as a community of dedicated experts, we can begin to address those challenges.

The IEEE Cybersecurity Initiative, chaired by Robert Cunningham and supported by Brian Kirk of the IEEE Computer Society, provided funds and organizational support that were essential to the success of the workshop. SRI International hosted the workshop and provided additional support through its Internet of Things Security and Privacy Center.

BACK TO TABLE OF CONTENTS

References

1. C.E. Landwehr, "A Building Code for Building Code: Putting What We Know Works to Work," *Proc. 29th Ann. Computer Security Applications Conf.*, 2013, pp. 139–147; www.landwehr.org/2013-12-cl-acsac-essay-bc.pdf.
2. T. Haigh and C.E. Landwehr, "Building Code for Medical Device Software Security," *IEEE Cybersecurity*, 2015; www.computer.org/cms/CYBSI/docs/BCMDSS.pdf.
3. M. Alt et al., "Design Flaws and Security Considerations for Telematics and Infotainment Systems," *IEEE Cybersecurity*, 2017; www.computer.org/cms/CYBSI/docs/CSD-telematics.pdf.
4. C.E. Landwehr and A. Valdes, "Building Code for Power System Software Security," 2017; www.computer.org/cms/CYBSI/docs/BCPSSS.pdf.
5. D.V. Gibson, G. Kozmetsky, and R.W. Smilor, eds., *The Technopolis Phenomenon: Smart Cities, Fast Systems, Global Networks*, Rowman & Littlefield, 1992.
6. H. Schaffers et al., "Smart Cities and the Future Internet: Towards Cooperation Frameworks for Open Innovation," *The Future Internet*, LNCS 6656, Springer, 2011, pp. 431–446.
7. A. Zanella et al., "Internet of Things for Smart Cities," *IEEE Internet of Things J.*, vol. 1, no. 1, 2014, pp. 22–32.
8. B. Schneier, "Security and the Internet of Things," *Schneier on Security*, blog, 1 Feb. 2017; www.schneier.com/blog/archives/2017/02/security_and_th.html.
9. US Nat'l Inst. of Science and Technology (NIST), *Security Considerations in the Information System Development Life Cycle*, NIST Special Publication 800-64, revision 2, 2008; http://csrc.nist.gov/publications/nistpubs/800-64/NIST-SP800-64.pdf.
10. J.H. Saltzer and M.D. Schroeder, "The Protection of Information in Computer Systems," *Proc. IEEE*, vol. 63, no. 9, 1975, pp. 1278–1308.
11. S. Bratus et al., "Beyond Planted Bugs in 'Trusting Trust': The Input-Processing Frontier," *IEEE Security & Privacy*, vol. 12, no. 1, 2014, pp. 83–87.
12. S.W. Smith and J.S. Erickson, "Never Mind Pearl Harbor—What about a Cyber Love Canal?" *IEEE Security & Privacy*, vol. 13, no. 2, 2015, pp. 94–98.

BACK TO TABLE OF CONTENTS

Workshop Participants

The following people participated in the IEEE Internet of Things Building Code for Building Code (IoT BC2) Workshop. Note that organizational affiliations are shown for information only. The workshop results and report don't necessarily represent the views of these organizations.

Gabriela Ciocarlie, SRI International
Camille Cobb, University of Washington
Bogdan Copos, SRI International
Robert Cunningham, MIT Lincoln Laboratory (Chair, IEEE Cybersecurity Initiative)
Sven Dietrich, The City University of New York John Jay College of Criminal Justice
Bryan K. Fite, BT Counterpane
Kate Garman, City of Kansas City, Missouri (now with the City of Seattle, Washington)
Merike Kaeo, Farsight Security
Brian Kirk, IEEE Computer Society
David Kravitz, DarkMatter LLC
Bhaskar Krishnamachari, University of Southern California
Carl Landwehr, George Washington University
Ulf Lindqvist, SRI International
Michael Locasto, SRI International
Daniel Mosse, University of Pittsburgh
Steve Myers, Indiana University Bloomington
Luis Paris, Centri
Atul Prakash, University of Michigan
Rei Safavi-Naini, University of Calgary
Devu Manikantan Shila, United Technologies Research Center
Sean Smith, Dartmouth College
Laura Tinnel, SRI International
Paul Vixie, Farsight Security
Tim Weisenberger, SAE
Jan Werner, University of North Carolina at Chapel Hill
Brook Wessel, Elliseaum International, LLC
Dongyan Xu, Purdue University
Qinqing (Christine) Zhang, United Technologies Research Center

BACK TO TABLE OF CONTENTS