IAB IOTSI Workshop
Internet-Draft
Intended status: Informational
Expires: August 24, 2016

H. Birkholz
Fraunhofer SIT
N. Cam-Winget
Cisco Systems
C. Bormann
Universitaet Bremen TZI

February 21, 2016

A Security Automation view of IoT Interoperability

Abstract

Security Automation poses a number of specific requirements on the information that IoT devices make available about themselves. In particular, security automation needs to operate on collections of rather heterogeneous devices in an integrated way. Well known principles, such as the differentiation of different planes of operation or the composition of devices — and now things — can be leveraged as guidance to achieve some of the necessary semantic interoperability. This short position paper discusses ways to achieve some of the necessary semantic interoperability.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

In the Internet today, many security processes are still manual to a surprising extent. IoT systems will need to scale beyond what humans can juggle by their own; this requires automation. Today, IoT protocols are proliferating, each often tailored to specific industry verticals and more, to specific tasks dedicated to direct machine-2-machine interaction. This diversity, coupled with the perceived need for quick completion of the relevant standards, poses great challenges in achieving semantic interoperability for a general IoT. However, all these devices must be managed cohesively and a common management layer could be a good focusing point to discuss semantic interoperability.

Traditionally, the management layer is also the starting point to facilitate security automation that attempts to automate the collection, transfer and processing of information required to perform security processes. In essence, effective security automation in a general IoT does depend on the intended semantic interoperability so that its broad deployment "would free security practitioners to focus on high priority tasks and should improve their ability to prioritize risk based on timely information about threats and vulnerabilities" [SACM-CHARTER].

One approach that has turned out to be a useful basis in this space is to collect metadata information about the assets that build up an information system (CISCO Identity Service Engine [ISE], MITRE/CIS [OVAL], [RFC5209], [I-D.ietf-sacm-architecture]). An approach that achieves this goal via semantic representation is the Interconnected-asset Ontology [IO], which provides a set of technology-agnostic concepts and their semantic relationships (including taxonomies) to represent interconnected network devices of different vendors that compose complex networks.

As the Internet of Things is trying to scale the Internet by a few more orders of magnitude, obviously, a number of the engineering trade-offs that apply to existing security automation scenarios may change with that additional scaling. General IoT devices may be more constrained, the networks may be more constrained, and the scaled-down overall cost of ownership per device may pose even more important constraints. At the edge to operational technologies, the lifecycle expectations change, and additional considerations beyond security, such as safety, may enter.

1.1. Terminology

Interoperability between devices of different vendors and different classes is a known requirement in the traditional Internet. A well-known challenge is the corresponding mapping of functions and function names across vendors and classes. Common tools to express this mapping are supported by ontologies. The T-Box [DL] — the terminological component — of an ontology can be used to represent a vocabulary of terms and their semantic relationships. If the T-Box is designed appropriately [ONTO101], a corresponding A-Box — the assertion component — can be used as the basis for the automation of security decisions by providing, for example, the known configuration and state of a swarm of IoT devices, safety requirements, or even data needed to understand the cost of ownership.

In order to give structure to the challenge of creating an all-encompassing T-Box for a general IoT, terminology from the traditional Internet can be leveraged to create three classes for

entities and interactions:

- the data plane functions that constitute the primary purpose the device incorporates,
- the control plane functions that enable a device to function continuously in concert and

alignment with other devices, and
- the management plane functions that enable the steering, configuration and monitoring of devices.

These three sets of entities, functions, and corresponding interactions continue to apply in the Internet of Things, and there is merit in identifying the functions and capabilities of IoT devices and assign a similar — or even identical — terminology to these that aligns with existing architectures.

2. Device Composition

In today’s complex systems, microcontroller subsystems that perform specific sensing or actuating services are often connected to a somewhat larger system creating a composite using a *local interconnect*, such as USB on the desktop or I2C within a larger sensing system. The subsystem microcontroller outsources several functions, such as network access, authorization, and other management functions to the larger system device it is a component of. One of the objectives of enabling constrained devices to directly connect to the Internet is to reduce the need for tying the operation of such subsystem microcontrollers to specific local interconnects, but this will not obviate the benefits provided by composition.

On the wireless side, the role of a local interconnect is often taken by more complex standards such as Bluetooth. This creates a much looser coupling in the composition that apportions more of a separate identity to the connected subsystem. In these approaches, the Internet access still is confined to the larger system in the composite, which typically also means the larger system needs to translate the management functions of the local interconnect to its external management plane interface.

In the network element space, it is quite usual for devices to merge their management functions and therefore become a single distinguishable device on the management plane, while the control functions of the actual devices remain separate to a degree to enable this kind of concerted behavior. In the IoT, constrained devices might not have enough resources to provide all management functions individually. Alternatively, a swarm of devices could be represented by a single entity on the management plane, orchestrated by the control functions of the swarm. The incentive to merge devices on the management plane might be different, but the result, the architecture can be very similar and therefore benefit from an aligned terminology.

The semantic representation used in the Interconnected-Asset Ontology acknowledges the fact that networks — which are the basis for complex composites in the IoT space — can themselves be complicated composites with multiple levels of layering, nesting or virtualization, each of them driven by different enablers and disablers of communication. There are, for example, multiple “views” of the same composite that is called a network, and in each of these “views” the term “reachability” can have a different meaning.

As a result, instead of trying to anticipate and therefore facilitate a specific use of the semantic representation, all general interconnected relationships are taken into account by representing the different properties of distinguishable components of a network in detail. Similarly, instead of providing information tailored for a specific security automation work flow that can be envisioned today, semantic interoperability should focus on providing information that can be used in today’s and tomorrow’s work flows.

A corollary of the above is that the description techniques also need to be composable. This is often somewhat more naturally the case for self-describing information sets, while metadata approaches sometimes tend to use more monolithic approaches (“schema languages”). The recent move of practical web technologies from the more integrated XML technologies to the more composable JSON approaches may be an expression of this, but today’s Web APIs often rely on a large amount of documentation, combined with implicit understanding, to be able to interpret and form API data.

3. The Need for Self-Description

In future IoT networks, one of the major challenges will be to simply keep up with the myriad of devices in the network and their security properties. The traditional approach, erecting a strong network perimeter and gating security on network access, will not work in the multi-stakeholder environment typical for IoT. Instead, the devices will need to exhibit more intrinsic security at the application level. Besides network management, the management of these devices will include setting up the application associations required, configuring authorization, and managing the security properties themselves such as assessing security posture and patch management.

For the latter, some form of information about the software that is running on the IoT device will be required. SWID Tags as standardized in [ISO-IEC-19770-2] and [NISTIR-8060] may provide an important source of structure for this information. Software Identifiers (SWID) help to “consistently and accurately identify software products that need to be managed for any purpose, such as inventory, licensing, cyber-security, or the management of software and software dependencies.” and detailed information provided by a [ISO-IEC-19770-2] SWID tag “can be used as part of an automated policy decision to allow or prevent the software installation”. As SWIDs are used today for larger devices, the prominent vulnerability assessment scenario [I-D.coffin-sacm-vuln-scenario] uses SWID tags as the semantic link between services (e.g. on control or management plane) and version specific vulnerabilities [VULN]. “Categories of tag consumers include software consumers, inventory/discovery tools, inventory-based cybersecurity tool providers (e.g., providers of software vulnerability management products, which rely on accurate inventory information to support accurate vulnerability assessment), and organizations that use these tools” [NISTIR-8060].

A composite device will need to make SWID tags available for all software running on itself and on its component devices. It will need to provide a combined view, but also needs to enable the management of its specific subsystems based on the set of software running on each of them. Individual SWID documents and their relationship between each other can help to represent this interconnected relationship of composite devices in a general IoT. The value of SWID tags in the context of the traditional, perimetered or gated Internet is apparent. Which subsets or additional characteristics will provide the same “semantic” glue in a representation of a general IoT is an open question.

In a REST environment, a SWID is an example of a non-trivial media type. While it is possible to deconstruct SWIDs into, say, linked data, governed by a specific ontology, it seems more productive to actually standardize SWIDs and their interchange format. Today, SWID is defined in a W3C XML Schema. A concise form of a SWID data structure may be more appropriate for a constrained IoT device.

4. Conclusion

The present position paper argues for equipping IoT devices with information they can make available to Security Automation processes, without necessarily trying to anticipate what those processes are going to be. This requires making available declarative (as opposed to procedure-oriented) information, taking composability of this information (possibly in parallel with composability of the devices themselves) into account. The paper does not go into detail on the authentication or authorization of that information, or on the way this information is conveyed (e.g., directly or by referral, in a push or pull interaction, or data formats). It identifies the SWID data structure as one structure that can be used to convey such information, and argues for defining appropriate Media Types for similar relatively complex types of security information. It also mentions the Interconnected-Asset Ontology (IO) as an ontology that has been designed with a strong view on composability and maintaining declarative information over several layers of abstraction.

5. Informative References

[DL] Baader, F., "The description logic handbook: Theory, implementation and applications", Cambridge university press, 2003.

[I-D.coffin-sacm-vuln-scenario] Coffin, C., Cheikes, B., Schmidt, C., Haynes, D., Fitzgerald-McKay, J. and D. Waltermire, "SACM Vulnerability Assessment Scenario", Internet-Draft draft-coffin-sacm-vuln-scenario-01, January 2016.

[I-D.ietf-sacm-architecture] Cam-Winget, N., Lorenzin, L., McDonald, I. and l. loxx@cisco.com, "Secure Automation and Continuous Monitoring (SACM) Architecture", Internet-Draft draft-ietf-sacm-architecture-05, October 2015.

[IO] Birkholz, H., Sieverdingbeck, I., Bormann, C. and K. Sohr, "IO: An interconnected asset ontology in support of information security applications", DOI 10.1109/ARES.2012.73, in 7th International Conference, Availability, Reliability and Security, Prague, Czech Republic, ISBN 978-1-4673-2244-7, 2012.

[ISE] Heary, J. and A. Woland, "Cisco ISE for BYOD and Secure Unified Access", Cisco Press, June 2013.

[ISO-IEC-19770-2] ISO/IEC JTC1, "Information technology -- Software asset management -- Part 2: Software identification tag", ISO/IEC 19770-2:2015, 2015.

[NISTIR-8060] Waltermire, D., Cheikes, B., Feldman, L. and G. Witte, "Guidelines for the Creation of Interoperable Software Identification (SWID) Tags (Final Public Draft)", December 2015.

[ONTO101] Noy, N. and D. McGuinness, "Ontology development 101: A guide to creating your first ontology", Stanford knowledge systems laboratory technical report KSL-01-05, Stanford medical informatics technical report SMI-2001-0880, 2001.

[OVAL] Mitre and Center for Internet Security (CIS), "Open Vulnerability and Assessment Language"

[RFC5209] Sangster, P., Khosravi, H., Mani, M., Narayan, K. and J. Tardo, "Network Endpoint Assessment (NEA): Overview and Requirements", RFC 5209, DOI 10.17487/RFC5209, June 2008.

[SACM-CHARTER] IETF, "Security Automation and Continuous Monitoring (sacm), Charter for Working Group", 2015.

[VULN] Martin, R., "Managing vulnerabilities in networked systems", IEEE Computer 34(11), pp. 32--38, 2001.

Authors' Addresses

Henk Birkholz
Fraunhofer SIT
Rheinstrasse 75
Darmstadt, 64295
Germany
EMail: henk.birkholz@sit.fraunhofer.de

Nancy Cam-Winget
Cisco Systems
3550 Cisco Way
San Jose, CA 95134
US
EMail: ncamwing@cisco.com

Carsten Bormann
Universitaet Bremen TZI
Postfach 330440
Bremen, D-28359
Germany
Phone: +49-421-218-63921
EMail: cabo@tzi.org