Carnegie Mellon University
Software Engineering Institute

# TWO PERSPECTIVES ON IOT SECURITY: STANDARDS AND RUNTIME ENFORCEMENT

*Grace A. Lewis*

March 2019

## Introduction

Internet of Things (IoT) devices are increasingly being used to support operations in tactical environments, such as those experienced by first responders, military, medics, and other field personnel. In addition to disconnected, intermittent, and limited (DIL) network connectivity, threats in these environments often include sabotage, capture, and impersonation, of both IoT devices and their clients. Therefore, strong yet decentralized authentication and authorization mechanisms are necessary to mitigate these threats. In addition, despite the Department of Defense (DoD's) current use of IoT devices in supervisory control and data acquisition (SCADA) systems, and its interest in using such devices in tactical systems, adoption of IoT by has been slow mainly due to security concerns (e.g., reported vulnerabilities, untrusted supply chains. At the same time, DoD recognizes the rapid pace at which the IoT commercial marketplace is evolving, and its urgency to embrace commodity technologies, to match its adversaries.

This white paper describes two projects related to IoT security. The first project, *Authentication and Authorization for IoT Devices in Tactical Environments*, is a collaboration with Authentication and Authorization for Constrained Environments (ACE), an active working group in the Internet Engineering Task Force (IETF). As part of this work we developed an implementation of ACE that includes capabilities for bootstrapping of credentials and token revocation to account for threats and characteristics of tactical environments.

However, although appropriate and relevant, using ACE for IoT security requires the fabrication of ACE-compliant IoT devices. As is common with standards, there is rarely a single standard, many standards come and go, and there are often not enough financial and technical incentives for manufacturers to commit to a standard.

Therefore, the second part of this white paper discusses the project *High-Assurance Software-Defined IoT Security*. The goal of this work is to move part of security enforcement to the network to enable the integration of IoT devices into systems, even if the IoT devices are not fully trusted or configurable, by creating an IoT security infrastructure that is provably resilient to a collection of prescribed threats.

## Authentication and Authorization for IoT Devices in Tactical Environments

The starting point for the standardization perspective on IoT security is Authentication and Authorization for Constrained Environments (ACE), an active working group in the Internet Engineering Task Force (IETF). The charter for this group is to develop a standardized solution for authentication and authorization of IoT devices by extending OAuth 2.0. The basic protocol flow for ACE is shown in Figure 1. A client C (e.g., mobile device) that wishes to access a resource at a Resource Server (RS) (e.g., IoT device) first contacts the Authorization Server (AS) to request an access token (1). The AS determines whether C has the right to the requested token, and if that is the case, creates and returns a token to C along with access information (2). The access information enables C to set up a secure connection with RS and to prove that it is the rightful owner of the access token (via a Proof-of-Possession (PoP) key). C then sends a resource request and the access token to the RS (3). After verifying that the token grants access to the requested resource and that C is the rightful owner of the token, RS responds to the request (4).

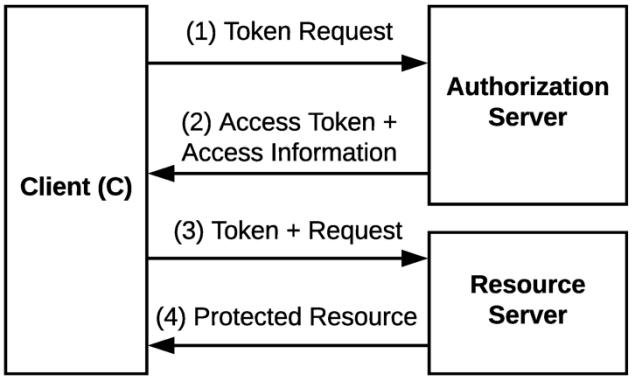


*Figure 1: ACE Basic Protocol Flow*

An analysis of ACE for use in tactical environments, done using threat modeling, identified two gaps:

1. Bootstrapping of credentials: Bootstrapping of credentials/keys is out of scope for ACE. ACE assumes that clients and RSs have previously securely exchanged credentials with the AS. This is not an oversight in ACE, but rather a consequence of the heterogeneity of IoT devices and pairing mechanisms. While acceptable in the context of home and potentially industrial environments, not including bootstrapping as an integral part of the process is a risk in tactical environments given that node capture and impersonation are high-impact, high-probability threats. We define bootstrapping as (1) granting clients and RSs access to the network, and (2) exchanging credentials to set. To address this gap, we developed a pairing mechanism for IoT devices that involves scanning QR codes as an out-of-band channel for exchanging initial encryption keys between IoT devices and the Authorization Server.

2. On-demand token revocation: Due to the often chaotic nature of tactical environments, it is important for clients and IoT devices to know if active tokens in their possession belong to compromised IoT devices and clients, respectively. Token revocation, other than time-based via expiration times associated to tokens, is not implemented in ACE. The problem is that (1) compromised clients will have access to resources until token expiration time, and (2) clients will have access to compromised resources until token expiration time. To address this gap, we developed a mechanism for periodic introspection between IoT devices and the AS, and between clients and the AS.

The resulting system was evaluated in terms of resource consumption via experimentation, against the threat model and using vulnerability analysis with simple attack trees. All code and supporting documentation are available at https://github.com/SEI-TTG/ace-client/wiki.

## High-Assurance Software-Defined IoT Security

As stated earlier, a standards-based approach such as the one described above, requires ACE-compliant devices, and therefore does not take advantage of opportunistic integration of novel IoT devices. To address this limitation, we propose to move part of security enforcement to the network to enable the integration of IoT devices into DoD systems, even if the IoT devices are not fully trusted or configurable.

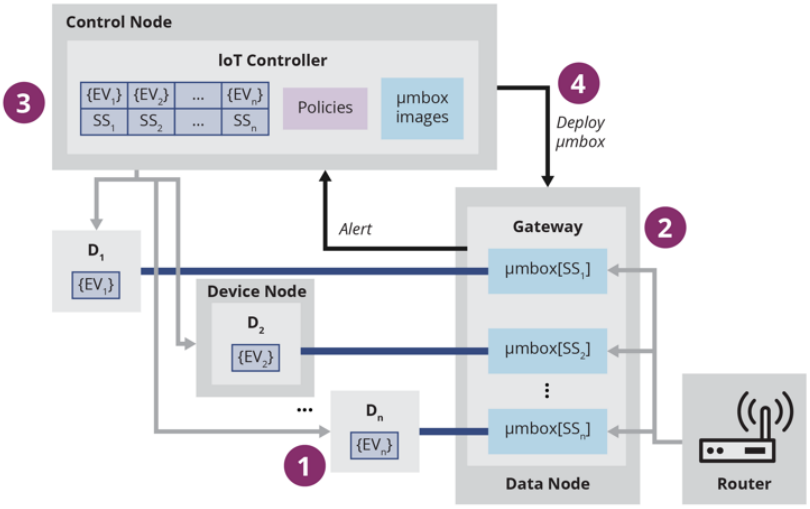


*Figure 2: IoT Security Infrastructure*

We are developing a high-assurance software-defined IoT security infrastructure that is provably resilient to a collection of prescribed threats. The software-defined aspect is the use of concepts and constructs from software-defined networking (SDN) and network function virtualization (NFV) to create a highly dynamic IoT security infrastructure, as shown in Figure 2. Each IoT device connected to the infrastructure senses or controls a set of environment variables called EV (1). The network traffic to and from each of these devices is tunneled through a µmbox, or micro middlebox, that implements the network defense for that device's current security state, which could be a firewall, an intrusion prevention system, or any combination of defenses (2). This is the data plane. The IoT controller, which is the core of the control plane, maintains a shared statespace composed of all the environment variables controlled or sensed by all IoT devices, as well as the security state for each device, which can be normal, suspicious, or under attack (3). The advantage of a shared statespace is that the controller can reason about the full set of variables and security states to account for implicit interactions via the physical world. Finally, changes in this shared statespace are evaluated by policies and may result in the deployment of new µmbox (4). For example, an alert from a µmbox, combined with a value currently being sensed by an IoT device, could lead the infrastructure, through policy evaluation, to determine that a certain IoT device is potentially being attacked (suspicious) and therefore it needs to deploy a new µmbox that defends against the potential attack. The high-

assurance aspect is the use of überSpark—a framework for building secure software stacks (https://uberspark.org/)—to incrementally develop and verify security properties of elements of the software-defined IoT security infrastructure. Examples of security properties for the control node are policy data integrity and µmbox image storage integrity. Examples for the data node include isolation between µmboxes of different trust levels and µmbox deploy-time integrity.

Artifacts that have been developed in the context of this project are

- threat model based on operation in edge environments
- architecture, design, and implementation of an initial version of the infrastructure
- IoT policy "data capsule" construct using überSpark that guarantees that (1) policy data can only be modified by prescribed policy interfaces and (2) policy interfaces are only available to authorized clients (policy data integrity)
- expressive policy model to define policies for IoT devices—a policy is defined as (1) the set of conditions that indicate a change in the security state of an IoT device and (2) the set of actions taken when the conditions are met
- dashboard for easy configuration and monitoring of the system

A live experiment will be conducted in summer 2019 to validate the resiliency and performance of the infrastructure.

## Contact Us

Software Engineering Institute
4500 Fifth Avenue, Pittsburgh, PA 15213-2612

**Phone**: 412/268.5800 | 888.201.4479
**Web**: www.sei.cmu.edu | www.cert.org
**Email**: info@sei.cmu.edu



This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM19-0281

.