View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Technical Disclosure Common

Technical Disclosure Commons

Defensive Publications Series

October 26, 2018

MANUFACTURER USAGE DESCRIPTIONS AND POLICY FOR INTERNET OF THINGS APPLICATIONS ON NON-DEDICATED HARDWARE

Chris Steck

Eliot Lear

Follow this and additional works at: https://www.tdcommons.org/dpubs_series

Recommended Citation

Steck, Chris and Lear, Eliot, "MANUFACTURER USAGE DESCRIPTIONS AND POLICY FOR INTERNET OF THINGS APPLICATIONS ON NON-DEDICATED HARDWARE", Technical Disclosure Commons, (October 26, 2018)
https://www.tdcommons.org/dpubs_series/1610

This work is licensed under a Creative Commons Attribution 4.0 License.

This Article is brought to you for free and open access by Technical Disclosure Commons. It has been accepted for inclusion in Defensive Publications Series by an authorized administrator of Technical Disclosure Commons.

MANUFACTURER USAGE DESCRIPTIONS AND POLICY FOR INTERNET OF THINGS APPLICATIONS ON NON-DEDICATED HARDWARE

AUTHORS:
Chris Steck
Eliot Lear

ABSTRACT

Techniques are described herein for extending Manufacturer Usage Descriptions (MUD) to onboard Internet of Things (IoT) applications on general purpose hardware in two ways that can work in tandem or separately. First, it allows an IoT application software package to securely present a MUD Uniform Resource Identifier (URI) at installation and/or first run, to the operating system on the local Personal Computer (PC), enabling the operating system to run the application in an appropriately restricted environment (e.g., a container, or Virtual Machine (VM) with its own Internet Protocol (IP) address). Second, it allows the network to then onboard the application in the restricted environment securely as a virtual IoT device on the network.

DETAILED DESCRIPTION

Techniques described herein cover the use case of creating a policy for an IoT application that is loaded onto a general purpose computing device (e.g., a Personal Computer (PC) with more resources available than required by the specific Internet of Things (IoT) application.

In this scenario, a Manufacturer Usage Descriptions (MUD) file from a PC manufacturer, if provided at all, would be relatively useless, necessarily requesting resources for everything the PC is capable of doing, because the manufacturer cannot know specifically what this device will be doing. This means the PC cannot help protect the network from the device. Conversely, utilizing the MUD file of an IoT application for the entire PC would restrict the PC's network usage by other applications/users adversely.

Not applying MUD at all would enable an IoT application, such as a smarthome management hub or a building automation control application, to be installed on that PC by the end user, and that application may have access to the PC and the entire network equivalent to that of the installing user's authorization level. There is presently no way for

Published by Technical Disclosure Commons, 2018

a standalone IoT application to tell the PC or network the exact resources it will need so both can restrict the application's access or otherwise take action to protect the network from the application running on the PC.

Techniques described herein extend MUD to onboard IoT applications on general purpose hardware in two ways that can work in tandem or separately. First, it allows an IoT application software package to securely present a MUD Uniform Resource Identifier (URI) at installation and/or first run, to the operating system on the local PC, enabling the operating system to run the application in an appropriately restricted environment (e.g., a container, or Virtual Machine (VM) with its own Internet Protocol (IP) address). Second, it allows the network to then onboard the application in the restricted environment securely as a virtual IoT device on the network.

First, the application installation process restricts the IoT application's resources. A secure, signed software installation package is executed containing the IoT application is initiated on an unconstrained, general computing device (e.g., a PC). Alternatively, the install package may contain only the URI to the manufacturer's application service, and the application may be first downloaded from there before the application itself can be installed.

During IoT application installation, the secure software installation package presents the URI pointing to a MUD file prepared by the software application manufacturer to the operating system's application registration/installation mechanism. Protection of the MUD URI inside the installation package is covered by normal secure software practices such as code signing. The local operating system installer may retrieve the application's MUD file.

The application installation process provisions only the resources the MUD file describes and completes installation. At least two options may be used. First, the installation process may optionally provision this combination of applications and resources as a VM with a unique IP and Media Access Control (MAC) address that appears as a unique virtual IoT device to the network. The VM may also be configured to constrain the local operating system resources available to the IoT application.

Second, the installation process may optionally provision this combination of applications and resources as a software container with a unique IP and MAC address that

will appear as a unique virtual IoT device to the network. The container may also be configured to constrain the local operating system resources available to the IoT application.

These two options present the IoT application to the network as a virtual IoT device and leave the policy enforcement to the network.

The installation process may create the virtual IoT device's container or VM image itself, or may be performed by the manufacturer's application service, and passed into the local installer to configure with an engine/hypervisor.

If a MUD file for the physical PC also exists, the operating system may combine the application MUD and device (PC) MUD files, but may not update the MUD file for the entire PC and instead associate that combined MUD file with only the virtual IoT device.

The result is that the installation process proactively restricts the resources available to the application based on the application's MUD file and, in the case of a container or VM, creates a virtual device with a unique identifier (e.g., an IPv6 address) for that combination of applications and resources running on a PC with otherwise unrestricted resources.

Techniques described herein also allow network onboarding of an IoT application on the PC. If the IoT application is presented as a secure virtual IoT device as created from a container or VM by the aforementioned process, the PC presents the virtual device with a unique IP address to the network for onboarding and a normal MUD onboard operation occurs with 802.1AR, where the application's MUD URI is passed to the network. The network may apply segmentation to the virtual IoT device, resulting in the PC and the virtual IoT device being in separate segments.

If the IoT application did not undergo the process to become a virtual IoT device the network may take restrictive policy actions such as not onboarding a device without a MUD file or not onboarding an "IoT device" with a MAC address previously known to the network (e.g., the PC's Network Interface Card (NIC)).

The result is that the network has proactively restricted the resources available to the application based on the application's MUD file on a PC with otherwise unrestricted resources.

Figure 1 below illustrates an application installation process for restricting an IoT application's resources by downloading the application wrapped in a virtual IoT device container or VM.

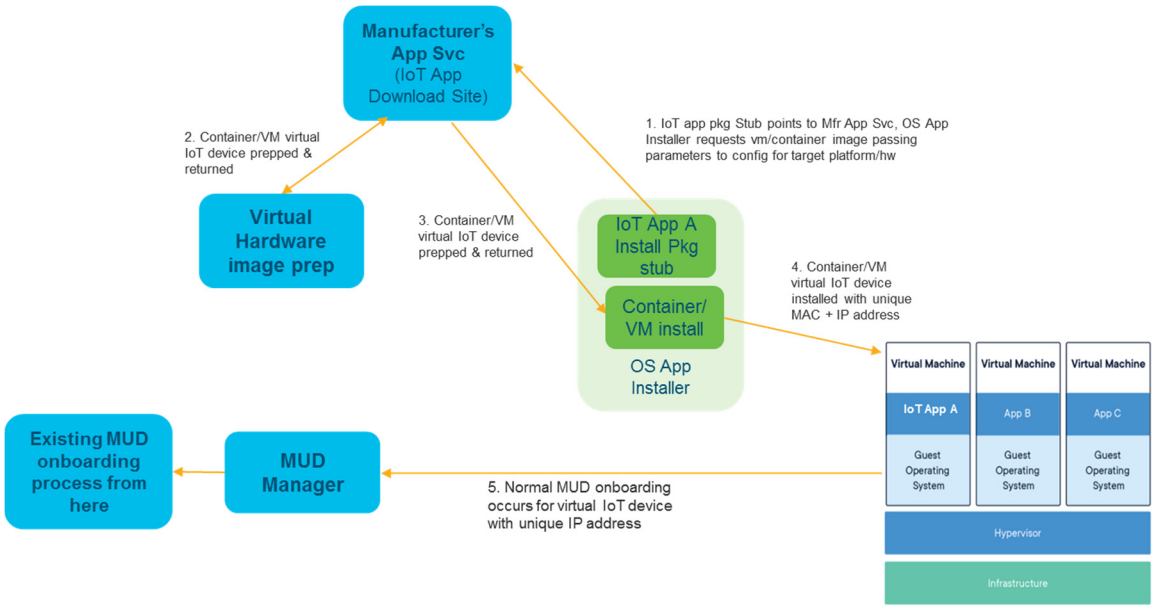

Figure 1

In summary, techniques are described herein for extending MUD to onboard IoT applications on general purpose hardware in two ways that can work in tandem or separately. First, it allows an IoT application software package to securely present a MUD URI at installation and/or first run, to the operating system on the local PC, enabling the operating system to run the application in an appropriately restricted environment (e.g., a container, or VM with its own IP address). Second, it allows the network to then onboard the application in the restricted environment securely as a virtual IoT device on the network.