AN MSO APPROACH TO SECURING THE 'INTERNET OF THINGS'

Jim Poder
IOT Strategy and Innovation

Abstract

As the Cable Industry prepares to support the onslaught of connected devices that make up the 'Internet of Things' we must consider our role in managing and securing the devices and services that appear on our networks and in our customer's homes. Traditionally the Cable Industry has provided these functions in other service areas through heavy specification, certification, and testing regimes, but in this new-world of devices that can not be the case. This paper describes how devices acquired through a variety of channels can successfully be managed and networks and services secured even without the traditional Cable model of specification, certification, and test. Through Device and service "fingerprinting" and taking an analytical approach to traffic monitoring, security can be delivered to the 'Internet of Things' world. This paper also explores the rapidly evolving world of 'Internet of Things' standards and discusses their relative merits where it comes to enabling the MSO to provide management and security while enabling service.

SCOPE OF THE IOT

The Internet of Things (IoT) is rapidly evolving! This is not much of a revelation to anyone who is involved in the tech industry and is not even to most technology users. Sensors and devices have become pervasive in our daily lives. From wearables like Fitbit or the Apple Watch to home automation and security systems like Xfinity Home and Lowes Iris to vertically integrated point solutions like the Nest Thermostat, many people are now living with something that could be described as the Internet of Things. Companies and market researchers will debate the size of the IoT, but most will admit that it is many billions of devices. Despite efforts toward creating and adopting standards like the AllSeen Alliance, Open Interconnect Consortium, Thread Alliance, and others, the "unified theory" of the IoT currently eludes us. There are many reasons why this is the case, but ecosystem control is one of the key drivers. Many device and sensor developers want a clear and easy path to consumers that doesn't leave them beholden to a service or platform provider. For them it can be an easier go-to-market strategy to build their own vertical integration where they build not only the device/sensor, but the communication protocol, data model, server, and web and mobile interfaces as well. The issue is that the power of the IoT is in correlating data from many devices and sensors to provide value to the user. Single data points don't allow for the rich set of services that have been envisioned. Integrated platforms and standards do not provide the panacea for devices either. Today the standards are incomplete, or under-supported, and often there are multiple standards that must be certified to in order to get devices deployed on a platform with a provider. Not only can this be very time consuming, but also very expensive for a smaller company.

Security is one of the largest and most complex issues when it comes to IoT. Systems typically have so many interfaces that it can be difficult to properly test them all to assure that there are no vulnerabilities. Additionally, the software and service layers evolve rapidly making it difficult to assure that new vulnerabilities are not introduced. We have seen many examples in the news over the past couple of years where systems

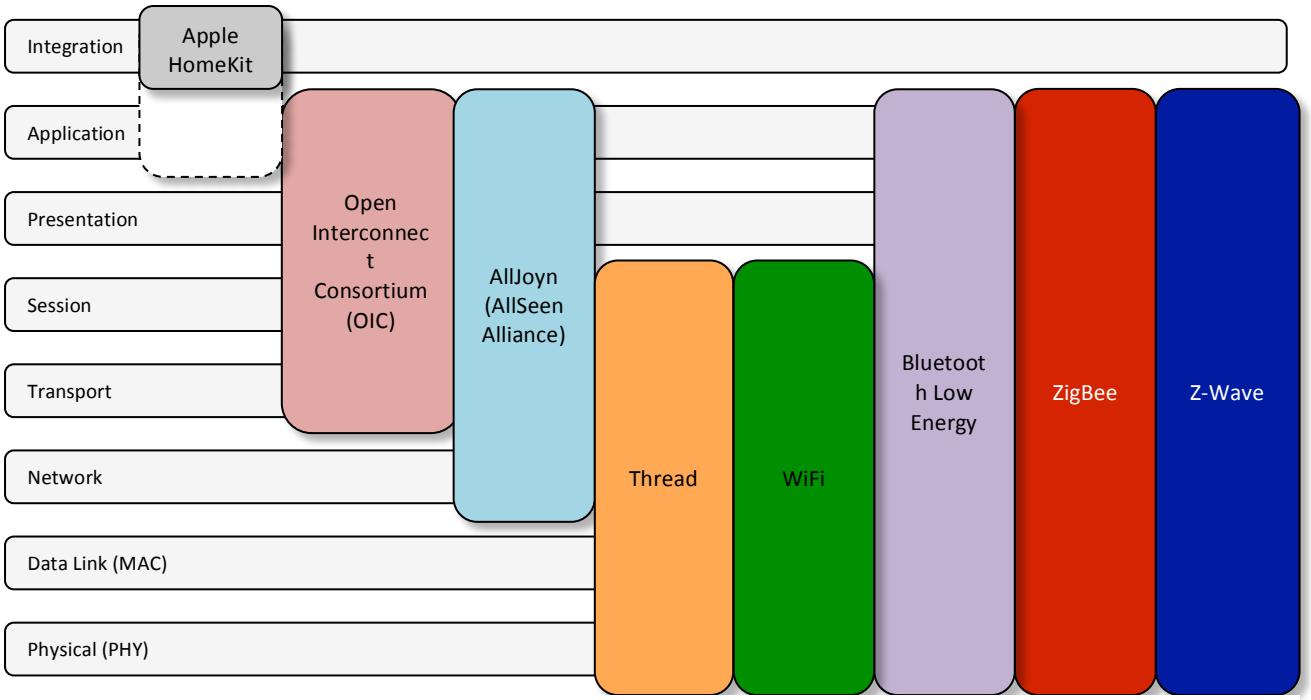

Figure 1: IoT Standards

have been exploited due to vulnerabilities of interfaces. Attacks can come from somewhat unrelated systems that happen to share common server or network elements. Several years ago there was an example of a large retailer in the US that had a significant theft of credit card data. It was discovered that the attack was perpetrated by infiltrating the climate control/HVAC systems for the stores and once the attacker had access to the internal network the Point-of-Sale (POS) system was open. While the POS system had strong security elements associated with it, the HVAC system did not. The moral to the story is that if security is not thought of and addressed holistically, the IoT cannot be made secure.

DEVICES OF THE IOT

As we look at the end devices of the IoT the challenges in securing them are numerous: They will come from a variety of suppliers, utilize a variety of standards and technologies. These devices may not be able to be truly authenticated, might not be trustworthy, and may not be patched or updated on a regular basis.

Role for Standards

There are many Standards being proposed to make up the IoT. Figure 1 shows each of their roles across the standard OSI stack. While these standards take good measure to ensure device authentication and secure communications, they fail to address the issues if the devices themselves have been comprimized.

The concern is that devices already on the home network, but have been comprimized can cause a serious issues such as sending data to non-authorized services or sending commands to other IoT devices.

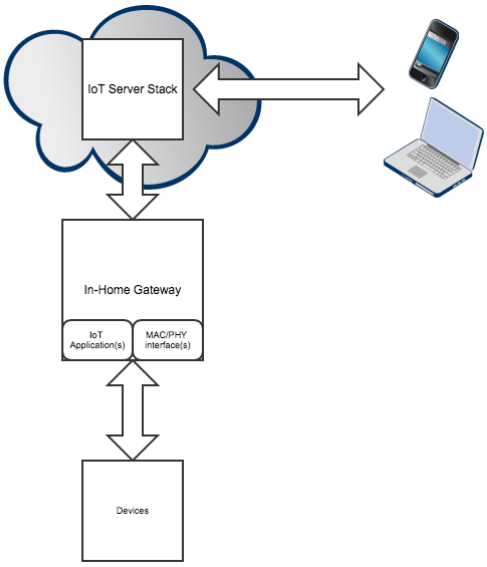

Figure 2 IoT Architecture

Figure 2 shows a high-level IoT architecture with devices communicating with servers/services by traversing the gateway and applications running therein. For devices utilizing TCP/IP protocols and a thin-gateway where the application is expected to reside in the IoT server Stack the device is left open to communicate with other servers. This communication may be part of normal, expected operations such as the device checking in with a manufacturer server to see if new firmware is available to be downloaded, but can also be used maliciously for a device to send data to other locations. Figure 3 describes the scenario where a rogue device (that is authticated to the network and is certified) is listening to messages on the device bus it can then forward these messages to another IoT server. Under most existing IoT standards this would be perfectly acceptable and compliant, but could result in a breach of personal information.

Rogue devices may also send commands to other devices on the network. A compromised device may send a command to a thermostat for instance to change the setpoint to 99 degrees! If the gateway is not blocking these sorts of malicious attacks it could cause real problems.

This really points out the need for device fingerprinting.

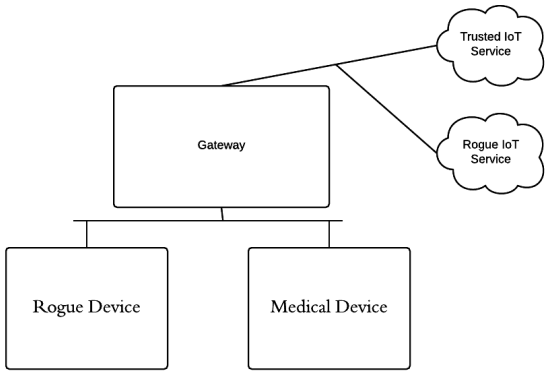

Figure 3: Rogue Device

DEVICE FINGERPRINTING

Device Fingerprinting can be thought of as network-wide pattern matching. When devices are communicating as expected the network runs fine, but should a device begin to operate outside of its expected parameters the network blocks the transactions and alerts are sent to the system and its users.

So how do we determine "expected parameters"?

White List, Black List and Grey List

When a device is trying to join the network and can be authenticated it will declare its capabilities as part of the authentication process (i.e. Thermostat, lighting controller, audio playback device, etc.) along with a manufacturer ID, device ID, hardware revision, and software revision. The gateway must then be responsible for doing a service level authentication for that device. Through a database stored on the gateway or service in the cloud the device will be determined to belong to a white-list of known and trusted devices, a black list of known untrusted devices, or a grey list of unknown or somewhat trusted devices. White list devices are those that have undergone rigorous testing and are certified throughout to be good

players. Black List devices are ones that have been proven to be bad players either through non-compliance with standards or expected operations or through rogue behaviors. These devices should not be allowed to participate in IoT services and the end user as well as the service provider should be warned of there presence so that they can be removed or mitigated. It is expected that the grey list of devices will prove to be the largest of the IoT. Devices that have not been seen before or that have not gone through the stringent certification process of white list devices will remain in this category. They do not have full permissions to access servers and services and any communications sent or received by these devices will be subject to fingerprinting to determine compliance with the security model.

Fingerprinting is the role of the gateway. During authentication and service level authentication the gateway will create a profile for each device containing device manufacturer, model, HW/SW revision. This will also contain expected behaviors of the device. These include:

- Heartbeat frequency, size of packet, and destination
- Normal communications frequency, size of packet, and destination(s)
- Firmware upgrade size and destination

The gateway keeps running counters for each of these attributes for each device on its network and will assure that

If a profile does not exist, the system will start with either a default model for that device type, or will begin the fingerprinting process. All of these are meant to establish what normal communication behavior means for a particular device. This may mean how often the device transmits a message, the destination for that message, or even the content of that message. For example, a connected thermostat may relay its status every 30 seconds, and send a heartbeat every 10 seconds and all of its communications have the climate control app within the gateway as its destination. If at some point the thermostat starts sending messages bound for a server on the internet or for a lighting control device in the home that communication is blocked and the system is notified of a potential issue. Given that there will be many manufacturer of similar devices, the fingerprinting algorithm will compare profiles of similar function devices against one another and should a device be sufficiently out different from its counterparts it will be flagged for possible blacklisting. One challenge to this model is that all intra-device communications inside the home must traverse the gateway where messages will be fingerprinted and evaluated as to their adherence to that device's allowed model before being delivered. This does not mesh with some IoT standards, notably Allseen which relies on peer-to-peer communications over the D-Bus architecture.[1] that rely on device-to-device communication using a publish-subscribe model on the home network.

Source of Rogue Devices

With the huge potential for suppliers for IoT devices it must be assumed that not all will be trustworthy, and some devices may come with software onboard that has alterior motives. This was seen in 2014 on some smart phones sold at retail in the U.S. that contained the malware 'DeathRing' preinstalled and note able to be uninstalled by the user.[2] This malware would transmit personal information from the smart phone to an unknown destination. This type of behavior must be expected, accounted for, and dealt with in the IoT. By fingerprinting devices and comparing device types this type of misbehavior can be rooted out.

Additional sources of Rogue devices involve long-lived devices that may not get maintained.

The Legacy of the IoT

Many of the smart home devices are ones that become part of the infrastructure of the home; Thermostats, outlets, lighting, appliances, motion detectors, door/window sensors, smoke alarms, etc. These devices are long-lived in homes and most consumers will never replace these devices once they are installed. This legacy-building problem is well known to MSOs. Take for instance the claim on an iconic lighting device of a 15 year life. Supporting it and insuring security robustness for that lifespan is going to be daunting. Feature upgrades aside, these devices are unlikely to have any long-term sustaining engineering support to patch bugs and security flaws that are found later in the device's life.

This presents a significant challenge to the smart home. Who is responsible for the security of the overall network and how will they accomplish this when all of the devices are not under their control?

Should a device manufacturer cease to exist or has declared end-of-life for a product there is the chance of a flaw being discovered later in that device's life that can be exploited maliciously. While these devices may be automatically added to the Black List and users notified, that is not a consumer friendly approach. Fingerprinting may help identify devices that have been exploited, but only after an attack has occurred and been identified. Fortunately, these attacks are limited to individual homes and are unlikely to be perpetrated on a wide-spread scale.

Another approach that is being addressed by Arm with their MBED OS is to require devices to run an underlying Operating System (OS) that can be upgraded by the system without harming the application software. [3]

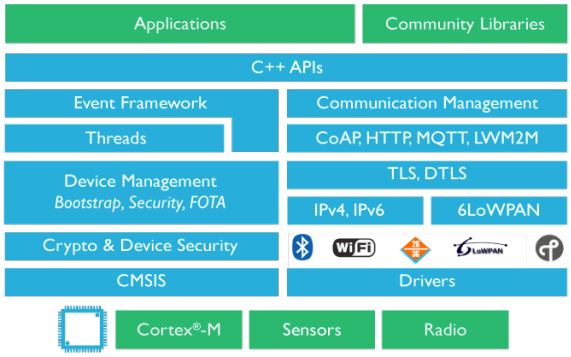

Figure 4: Arm MBED OS software Stack

Using this model, the security features and communications protocols can be upgraded without the need for the manufacturer's intervention. This can be difficult for all devices as many of the single-function, battery powered devices today do not run any sort of OS. As the need for manageability as well as over-the-air software upgrades becomes standard, it is believed that this will change.

CONCLUSIONS

Although it is too early to say which standards and approaches will prevail, it is clear that standardization and open security models are going to be an important piece of the IoT. Security of device and sensor networks must be thought of on a system wide level through device and communication fingerprinting along with dynamic white, black, and grey lists of accepted devices. These must not become an impediment to innovation and adoption, however. Service providers will have to be diligent in selecting architectures and devices, but must realize that they are building a legacy with consumers that will have to be supported for a very long time. We can then begin to realize the promise of services and analytics that can yield tremendous new businesses and services.

[1] Allseen Alliance Security https://allseenalliance.org/developers/learn/core/system-description/alljoyn-security

[2] Security Watch, PC Magazine December 8, 2014 http://securitywatch.pcmag.com/security-software/330164-mobile-threat-monday-deathring-malware-pre-loaded-on-android-smartphones

[3] Arm IOT device Platform http://www.arm.com/products/internet-of-things-solutions/mbed-IoT-device-platform.php