Comments to the Draft OTA IoT Trust Framework

Submitted by:
Beau Woods
Independent Security Professional
Stratigos Security

Overall comments:

Medical Devices

The definition of medical devices is still not yet firmly established in legal, regulatory, and common usage. It is likely that over the next few years the concept – and definition – will continue to be fluid as new device classes, use cases, and integrations emerge. To some degree it's irrelevant whether a device is classed a "medical device" or not when following a set of guidelines unless industry or governmental regulations require it. So perhaps a strict scoping is unnecessary except to define the thinking behind the creation of the framework.

System Boundaries

It may be of use to mention the concept of system boundaries. IoT devices deliver value by interacting with many other networks, devices, servers and other assets. Each device must define its own security perimeter, yet the security is linked to that of other components within the broader system. If one device takes input from a sensor network over Bluetooth, and stores data on a cloud server, what is the scope of the whole system, and what are the assumptions made about its boundaries? NIST 800-series documentation has much more discussion.

Supply Chain Rigor

Devices contain many different components, often from dozens of suppliers and sub-suppliers. Tracing the manufacture and provenance of these components assures that safe operating conditions are well understood – and that failure modes can be anticipated. These concepts are well known to traditional manufacturers and retailers, and should also be applied to software and hardware in IoT devices.

Tracing software and hardware gives a known, verifiable set of components. This list makes quality assurance and maintenance much easier. Requirements defined can be quickly validated by looking at the capabilities of the capabilities in the manifest. Software and hardware components with known flaws or defects can be found quickly and treated – before sale and during the operational lifetime of the device. Open source projects or commercial entities that go out of business can be identified and alternate plans for support developed. And potential licensing issues can be found to reduce legal threats to the business.

Verification Method

The framework validation method can work to improve the trustworthiness of IoT device vendors who adhere to it. Differences in implementation among various vendors and devices can establish a market advantage and provide a greater source of information to the buying

public if they are publicly documented. Full detail need not be provided – that can be saved for internal documents, cost advantage, and full auditing. But providing an overview of why and how each device addresses each requirement gives the buying public access to more information for decision-making. This method can also generate an industry consensus around certain security capabilities, yet is flexible enough to change as the marketplace evolves.

Specific Comments

Item #9

This item will hinder security researchers' ability to validate the secure operation of IoT devices. The U.S. Digital Millennium Copyright Act (DMCA) makes circumvention of technical protection mechanisms a federal crime with harsh punishments. Even if the device maker gives its permission for independent researchers to investigate security flaws (through a disclosure policy or a bug bounty), the activity may still be illegal. The OTA may wish to work with legal counsel on this point to ensure they are not inadvertently reducing the security of devices.

Item #12

The concept of adversarial resilience testing and threat modeling can improve this item. Many manufacturers see the software development lifecycle as a way to ensure devices perform as they are expected. However, they don't tend to check whether the devices can be made to do something they aren't expected. Manufacturers are typically good at anticipating threats from the physical environment, such as temperature, water, etc. But they often overlook the role that hostile and hazardous interactions can play over the Internet or other networks.

Penetration testing isn't sufficient. Pentesting also doesn't reliably find as many flaws as does the same type of adversarial resilience testing performed throughout the device lifecycle. It comes after the device has been designed and built, so findings at this stage definitely delay getting the product out, and increases costs versus finding flaws earlier.