KEY SECURITY CONSIDERATIONS FOR END-TO-END IOT PLATFORMS

To ensure data security in IoT we need to secure all the elements in the end-to-end chain, which includes the end-points, hardware and software interfaces, gateways, and everything that is a part of the local or remote cloud.

Securing things. Everything. Connected to the Internet. Simple enough, right?

IoT security is a vast topic, with a lot of research going on, a lot of literature available, and a lot of different viewpoints and arguments being presented. Within the topic, security needs and priorities depend on the segment and use case. For example, the security requirements in an industrial IoT setting are very different than those in health-care or home automation or infrastructure management. Our goal here is not to look at every aspect of IoT security, however tempting it may be. Instead, I will focus on a smaller subset: the most common factors that are relevant to the end-to-end IoT platform. It must be emphasized that security needs to be built into the architecture right from the design stage, and not as an afterthought. (In other words, do as we say, not as we did.)

Let us start with the basic questions: what are we trying to protect, from whom, and how?

Clearly, we are trying to protect the data: from someone stealing it, modifying it, and misusing it. We are trying to guard against someone working to affect the data generation and transportation across the pipeline. Protecting the volume, velocity, and veracity of the data is important because IoT systems are designed for interacting with and affecting the physical world, and also because of the (near) real-time nature of the data. In short, we are trying to ensure data security, data integrity, and uninterrupted data flow.

In order to accomplish this, we need to secure all the elements in the end-to-end chain, which includes the end-points, hardware and software interfaces, gateways, and everything that is a part of the local or remote cloud. This involves physical and network security of the infrastructure, access control, and identity management mechanisms, etc. We will assume that the physical security of the general infrastructure is taken care of.

Securing the end points

Let us start at the very beginning of the chain: the end-points (sensors and actuators, along with their interfaces). The wide variety of the end-points combined with various
interface technologies generates diversity and complexity when it comes to securing them.

The physical security of the endpoints could be a significant challenge. The usual mechanisms like physically isolated premises, locked rooms, etc. may not be feasible when one considers use cases that involve one or more of the following situations: a large number of sensors, deployment over a wide geography, or non-stationary (mobile) sensors. In all cases, the endpoints must be guarded against tampering or spoofing (whether malicious or innocuous). This may require non-traditional measures, like using tamper-proof enclosures, having the ability to monitor tampering attempts, sending alarms or disabling endpoints on such incidents, etc. In addition, periodic checks on the endpoint (heartbeat or ping) along with diagnostic test-run or collecting management information are helpful. While this may complicate the endpoint design, add overall system overheads, and increase costs, it will provide better security. Since not every deployment may need all of this, the actual implementation should be dictated by a cost-benefit-risk analysis of omitting certain security features in a particular use case.

Next to the physical security of endpoints, electronic communication between the endpoints and interfaces, gateways, etc. needs to be secured. Here again, the wide variety of connecting mechanisms - wired (analog or digital), Wi-Fi (802.11), Zigbee, Bluetooth (802.15.4) - pose a challenge, as one solution does not fit all cases.

Key security threats in IoT

Two important issues related to the endpoint communication are: (a) preventing someone from listening in and copying data, and (b) preventing someone from modifying the data. The first refers to the data security and the second refers to data integrity. Most of the wireless technologies under consideration here have strong encryption mechanisms for data security and integrity along with key management facilities. The security implementations span across all layers right from the MAC (Media and Access Control) to the application layer.

But the history of security tells us that nothing is foolproof. So what happens if someone still manages to get access to an endpoint, say a sensor and make it send incorrect data? There is no easy way to guard against such an attack. (It is assumed that physical access mechanisms have been breached by an attacker to reach this stage.) It is possible to detect such an attack by correlating the data from other sensors, determining possible abnormalities, and flagging the set of sensors as compromised or malfunctioning. In a way, devising such mechanisms can enable the system to diagnose and defend itself.

For wireless sensors, guarding against volume and velocity attacks is important. For example, in a Zigbee network, it is possible to launch attacks like flooding the wireless network, or to create strong interference on channels in use, thereby forcing retransmissions and ultimately reducing data volume and velocity. Guarding against such attacks, especially something that amounts to network jamming, is very difficult. Some of the wireless technologies employ channel hopping mechanisms that make them more resilient, but there is no foolproof way.

No fool-proof solutions

The next hop in our architecture is the interface device controller and the gateway. For many setups, these could be combined into a single physical implementation, with a reasonably powerful processor and operating system. This is the point that bridges the two worlds— the 'inside' (a home or a factory) and the 'outside' (the cloud). This is potentially one of the most vulnerable points, and strong security measures need to be put in place here. Using processors with secure boot and hardware security implementations are the best defense, since beating hardware-based security is very difficult. In addition, the usual measures like hardening the OS, access controls, and firewalls are necessary.

Beyond this point in the end-to-end chain, the rest of the network infrastructure is the usual, known territory. The security requirements and implementations are fairly standard and well-established, with firewalls, intrusion detection and prevention systems, endpoint protection systems, resiliency, high availability, etc. For the sake of brevity and avoiding repetition, we will not get into the discussion of these mechanisms.

"Security is imperfect: Recognize that, despite best efforts, it is quite likely that your systems will be breached and some user data compromised. Create explicitly written data retention policies as well as data breach disclosure and end-user notification processes."

The scope and implications of these policies and processes are even broader and more involved for IoT systems than for enterprise IT, which makes it all the more important that they be considered and created in advance.

In the end, there are no silver bullets for security, whether in enterprise IT or in IoT. The best systems will be those that are built on adaptive principles stemming from biology: design the best system given the known requirements about the environment; recognize that these requirements may be imperfect, incomplete, and susceptible to change; and build into the design mechanisms to monitor, repair, and work around broken assumptions.