www.pipelinepub.com
Volume 17, Issue 1

The Future of IoT: Secure by Design

By: Rob Spiger

IoT security is becoming an imperative. By 2021, it is forecast that 35 billion Internet of Things (IoT) devices will be installed worldwide, a number expected to grow to over 75 billion by 2025, according to Security Today. The IoT revolution will increase the number of computing devices by orders of magnitude. However, these devices will be built from the same imperfect software that we use today, and manual remediation will be much less practical or even unfeasible due to devices being too numerous, too inaccessible, or simply lacking a suitable interface.

As people increasingly rely on connected devices to make their everyday lives easier, it is imperative that device manufacturers and architects incorporate a security by design approach to protect the device throughout its whole lifecycle, from conception right through to the end of its lifetime. If security is an afterthought for developers, the device presents a vulnerable point for hackers to access or tamper with large amounts of personal or operational data being processed by the device and shared with the cloud. The impact of such a vulnerability can be hugely detrimental.

Security should not be an afterthought

All Internet-connected devices should be designed to protect themselves against network-based attacks. As such, device vendors must employ a wide range of hardware and software-based protection technologies to keep devices secure. Unfortunately, bugs and misconfigurations still lead to damaging exploits despite this. Furthermore, recovering a badly compromised computing device today usually involves manual intervention. For example, a new firmware or operating system must be loaded from an external storage device or a second computer before then being

re-joined to network services using passwords or other credentials, often under conditions of physical security.

Technologies that support reliable and secure remote computer management and recovery are already available for more costly devices. For example, service processors or baseboard management controllers (BMCs) are employed to manage desktops and servers, and intelligent backplanes are used to manage blades in data centers. However, these technologies are either unsuitable or inefficient for IoT due to their cost, form factors, power needs, or the lack of an out-of-band management channel.

A clear baseline for security is crucial

For devices to be secured from the start, it is imperative that developers have a robust starting point to work from. With many complexities and vulnerabilities within IoT devices, it is essential to have the ability to identify where these vulnerabilities are and a foundation for understanding how they can be best safeguarded.

The National Institute of Standards and Technology (NIST) is ensuring that engineers have the best tools to support the resilience of platforms against potentially destructive attacks with the three principles stated in its Platform Firmware Resiliency Guidelines (NIST SP 800-193). It outlines a collection of fundamental hardware and firmware components needed to boot and operate a system to protect the platform against unauthorized changes, detect unauthorized modifications that occur and recover from attacks rapidly and securely.

Within the protection principle, the guidelines outline mechanisms for ensuring that platform firmware code and critical data remain in a state of integrity and are protected from corruption, such as the process for ensuring the authenticity and integrity of firmware updates. The document also defines mechanisms for detecting when platform firmware code and critical data have been corrupted, leading to the recovery principle. During this process, the guidelines summarize the mechanisms for restoring platform firmware code and critical data to a state of integrity in the event that they are detected to have been corrupted, or when forced to recover through an authorized mechanism. The recovery aspect is limited to the ability to recover firmware code and critical data.

New tools for in-depth defense

This standard provides a set of baseline security provisions for all consumer IoT devices. It is intended to be complemented by other standards, defining more specific provisions and requirements for testing and full verification, such as the principles and technologies set out by the Trusted Computing Group's (TCG) upcoming Cyber Resilient Module and Building Block Requirements specification.

This specification defines a minimal set of hardware and firmware capabilities or mechanisms that enable cyber-resilient devices to be built, even at the lowest end of the cost, performance and complexity spectrum. This includes IoT devices and microcontrollers used in a wide range of applications. It also supports more complex devices by providing resilient capabilities to

subcomponents of devices that may have their own computing resources, critical firmware and critical data.

Defining new terms

New terms for primary architectural elements are defined. These include a resilience target, which can be recovered; a resilience engine that performs recovery actions; and a resilient authority that sets policies for recovery and initiates new actions when they are needed to put the resilience target in a trusted state.

A resilience target is a component that has mutable code and configuration information, whether firmware, software or both. By defining a resilience target within a device, the architect is able to draw a boundary around the code, configuration and runtime environment that other components in the specification can fix or service when a compromise occurs or patches need to be deployed due to a known vulnerability. Because it is anticipated that a resilience target might be compromised by malware, the architecture for recovery assumes the resilience target will not assist in the recovery processes. In fact, malware that compromises the resilience target may actively try to prevent recovery from occurring.

The resilience engine is a component that can service one or more resilience targets, even when they are uncooperative. With the ability to function even if the remote network is unavailable, the resilience engine supports configurable policies regarding when and how it performs servicing actions. To be able to respond to future circumstances not foreseen at the time the device was created, the resilience engine is expected to have the capability to receive new instructions from an authorized entity called a resilience authority.

Three important resilience building blocks enable the resilience engine to reliably service the resilience target. The first is a secure execution environment that the allows the resilience engine to run without interference from the resilience target. The simplest example of a secure execution environment is rebooting a device and having the resilience engine start first in the boot sequence and having the resilience engine decide when the resilience target starts. The resilience target can’t interfere with the resilience engine if it isn’t running.

The second resilience building block is a storage protection latch. Storage protection latches are intended to provide read and write protection for persistent storage. Initially after a device is turned on or restarted, the storage is accessible for reading and writing. Once a storage protection latch is enabled, it prevents reading, writing or both on an area of persistent storage. Storage protected by a storage protection latch provides an ideal location for a resilience engine to store is code and configuration. When the resilience engine runs first during boot, it can read and modify its persistent storage, but before it starts the resilience target, it can switch on the storage protection latch so the resilience target cannot modify storage used by the resilience engine.

The third and final resilience building block is a watchdog counter. The watchdog counter is how the resilience engine gains control to do servicing actions even if the resilience target is not cooperative. The simplest example is a “latchable” watchdog counter. It can be configured by the

resilience engine to unequivocally restart the platform after a fixed period of time, for example once a day. The restart gives the resilience engine a chance to do servicing actions. Obviously for some use cases, restarting a platform could be disruptive for the user, so the specification has other types of watchdog counters to address different situations.

Cyberattacks are evolving and continually improving. This means that solutions believed to be secure at the time of manufacturing are consistently in need of patches for unanticipated vulnerabilities or mistakes during the lifecycle of devices. By adopting the new cyber resilient building blocks, there can be strong protections for the resilience engine and reliable servicing chances when vulnerabilities or compromises of the resilience target need to be rectified. For a solution to be secure over time, robust building blocks like these mentioned above as well as active support is needed.

Preparing for unknown eventualities

In order to protect the ever-increasing number of IoT devices, the capabilities must be designed to be both simple to implement in hardware or firmware and simple to use for software. Such simplicity decreases the vulnerability of the security-critical firmware that operates them, while also minimizing cost, power consumption and size of the hardware.

By adopting and implementing these building blocks, device architects have a robust starting point to work from to ensure the resilience of any IoT devices, both now and in the future. As technology advances, new cyberattacks and vulnerabilities will arise that threaten the security of the growing number of IoT devices. However, through the development of specific use cases that require this new specification to ensure device security, manufacturers will be able to implement the particular building blocks required for their devices and the applications in which they will be used. Doing so safeguards IoT devices throughout their lifecycle, regardless of the sophistication of the potential attacks.