Can The Internet Of Things Be Secure?

William Wong

Tue, 2014-06-10 10:09

The Internet of Things is upon us. Whether it is secure is another matter.

Interest in securing networked digital devices has been on the rise. Yet it has become more important as security-related technologies like SSL have been breached. Of course, I'm referring to the Heartbleed bug *(see "*What Heartbleed Should Teach Embedded Programmers*" on electronicdesign.com)*.

Related Articles

How Nimble Is Your IoT Approach?

Understanding Wireless Routing For IoT Networks

Who's Doing What In IoT And M2M?

Interview: Dave Friedman Talks About How IoT Cloud Platforms Help to Accelerate Connected Product Delivery

What Heartbleed Should Teach Embedded Programmers

The Heartbleed problem is related to OpenSSL, which is an open source project that implements SSL/TLS. The security breach was due to a coding error and not in the protocol itself. It allowed an attacker to farm random data with the eventual possibility of obtaining information that could compromise the system or its users.

OpenSSL is not the only SSL/TLS solution available. There are dozens, including many open-source projects like GnuTLS and LibreSSL. Commercial implementations abound as well. Vendors have a vested interest in making sure their products are bug-free and secure. There are typically financial repercussions if a vendor's system is compromised, so this is one advantage to using a commercial solution. There is not a distinction between commercial and open source, though. A product can be both, although many commercial products tend to be closed-source solutions.

Commercial solutions often match components within the system. For example, HCC Embedded's MISRA-compliant TCP/IP has SSL/TLS support. MISRA is often used in automotive environments, but it has been used in a host of non-automotive embedded application arenas. The problem is that OpenSSL is used in a lot of open-source distributions and systems including much of the Internet. Fixes were forthcoming and the change was minor, but even now there are systems that still employ OpenSSL that is not patched.

The challenge for developers is that this type of change needs to be done in the field. It becomes even harder when the software is found in embedded devices. In one sense, embedded systems are at an advantage since they are typically designed to be updated remotely. The downside is that users can rarely instigate or manage these updates themselves.

Embedded devices like these are all around us, and the number is growing. They are in cars, televisions, and even thermostats like the Nest Learning Thermostat *(see "An Elegant Thermostat Designed For The Internet" on electronicdesign.com)*.

So will anything be really secure? Probably not, but brute force attacks can be made to take a very long time—like when the heat-death of the universe occurs. Alternatively, attack detection can be used to eliminate the target as with Apricorn's latest Aegis Padlock DT drive *(Fig. 1)*. This FIPS-2 (Federal Information Processing Standard) standard USB 3.0 drive is available in capacities up to 6 Tbytes. It is a handful, unlike some small, portable, and secure solid-state drives (SSDs) I have looked at *(see "Traveling With Portable Secure Storage" on electronicdesign.com)*. One thing they have in common is that they become forgetful if attacked repeatedly.

Of course, Apricorn's devices do not have to trash the actual data since the drives are encrypted. They simply destroy the encryption keys. Apricorn's drives have administrative and user passwords proving more flexible access management, but they are easily contained in the system's secure processor.

Secure Processors

Really secure systems start with a very secure root. TPM (Trusted Platform Module) is still an option on many motherboards that provide the starting point for a secure boot service. The problem is that TPM is outside of the main processor it is supposed to help protect. This is why many secure platforms incorporate security processors on chip and then try to prevent overt hacking by physically and electrically protecting the device. Secure processors are found in smart credit cards, and the same technology is becoming more common in general.

This same approach is used with the a range of secure processors available for application areas ranging from networking

to automotive. Typically, a main processor or multiprocessing system is paired with a security processor designed to handle the boot process and manage secure keys that will be used by the system. More general platforms are joining the specialized secure systems.

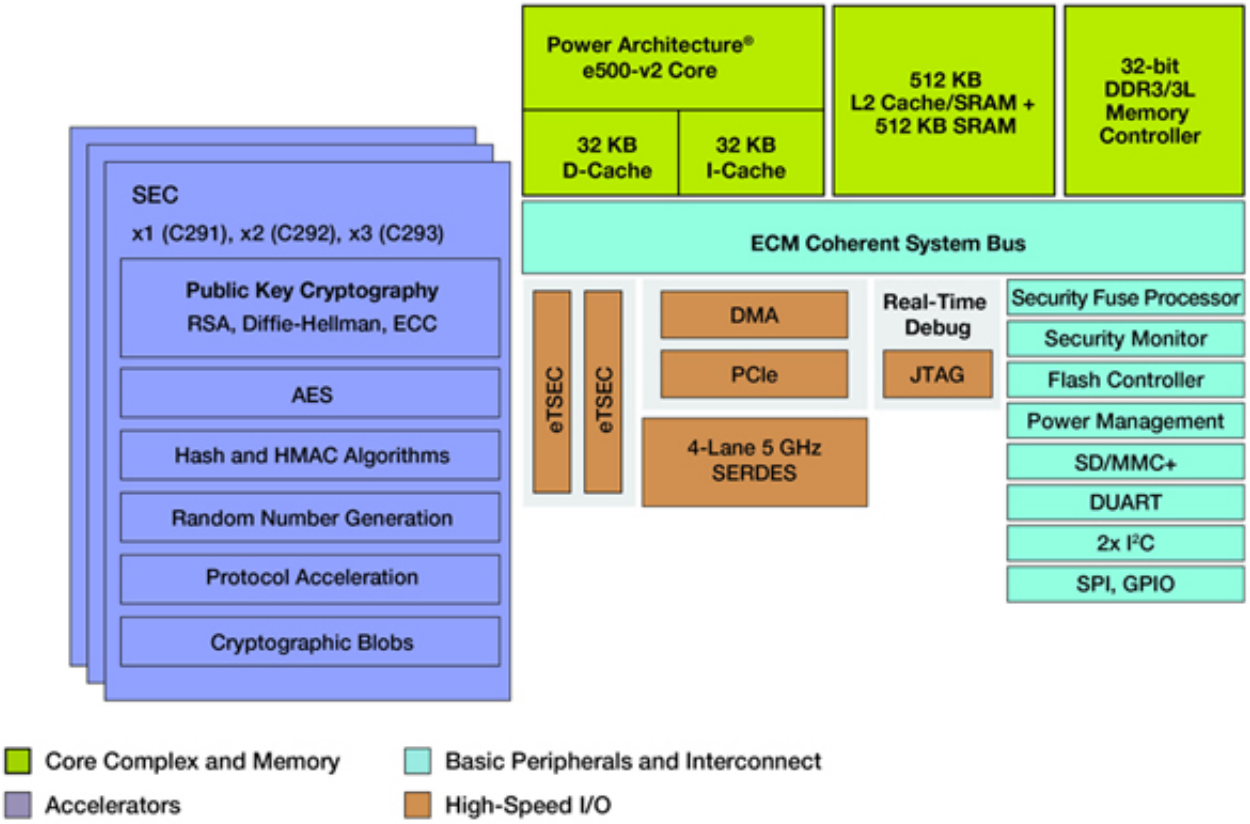
For example, AMD's Platform Security Processor (PSP) is an ARM Cortex-A5 tasked with securing the x86 side of things in multicore Mullins and Beema processors *(see "Platform Security Processor Protects Low-Power APUs" on electronicdesign.com)*. The PSP starts up first and enables the main cores to boot securely. It contains the keys to authenticate the code that will be used by the main host cores, allowing the operation of secure versions of operating systems like Windows 8 and Linux to run on these platforms.

The performance capabilities of many of the embedded security processors are what's interesting about the PSP. The Cortex-A5 is at the low end of the family, but it is more powerful than most of the Cortex-M series.

Users of AMD's chips will likely never know about the PSP, and programmers are unlikely to deal with it directly. This is one reason that the type of core used for the PSP is not relevant to the general user community. Still, it is an important part of the system, and it will be critical for anyone who wants a secure system.

Security processors tend to have a large role at boot time, but they can also be used during system operation. For example, their crypto hardware may be faster than the host, in which case the host may offload some of these chores. The security processor also may be involved in secure transactions with remote sites performing actions such as secure authentication before an update is applied to the overall system. It might also provide secure storage for keys used for digital rights management (DRM).

Secure processors should not be confused with crypto coprocessors like Freescale's C29x *(Fig. 2)*. These coprocessors are designed to augment throughput of secure network or storage traffic. They provide accelerators for processing stream data faster than a conventional processor could. This type of feature is sometimes built into a host, but they tend to be special-purpose devices. With this approach, a conventional host can be utilized since security is usually only a feature of the system, not the end solution.



Security has always been a software chore. It is common now in almost all computing devices as noted by the use of SSL/TLS. But that is just one aspect of security within a larger context that needs to start when a system boots. It must continue through a system's lifetime, including the update process.

Source URL: http://electronicdesign.com/dev-tools/can-internet-things-be-secure