Isolate Software Execution with a DeepCover Security Framework

A Typical Attack Scenario

In a typical scenario of a remote software attack on an IoT device—similar to the recent Mirai attack—a weakness in a communication stack is exploited, such as a buffer overflow, a lack of input data check, or a hard-coded password. A successful attack leads to the injection of code into the targeted device or micro.

Once the injected code is within the platform, it is executed, takes full control of the software and, ultimately, takes full control of the device. The consequences of losing full control of the device depend on the device's context; the result could be an exfiltration of sensitive data and code, or the device could be turned into a zombie to become part of a botnet (Figure 1).

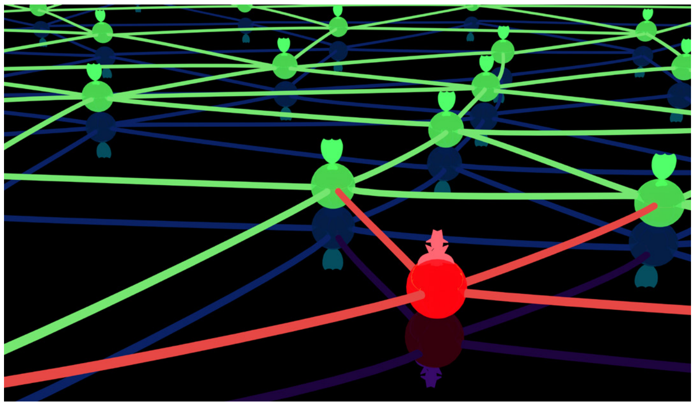
Figure 1. Depiction of a Botnet

The Typical Protection Strategy

The protection approach taken against this type of attack is usually based on a Memory Management Unit (MMU), a hardware block available on large applications processors that use microcontroller cores such as the Arm® Cortex®-A family. These solutions employ isolation software, typically called hypervisors, that use the MMU to define distinct and separate software "boxes" or "containers."

The hypervisor's main objective is to define the resources to be allocated to each of the software boxes and to determine a set of exclusive communication channels between them. Once defined, it is the hypervisor's duty to provide control and guarantee that the communication rules are always correctly followed. Examples of actions that might be detected and rejected by the hypervisor include attempts to obtain additional resources or to perform an illegal or unauthorized access. All software boxes must be executed within their own boundaries and cannot exceed their privileges.

The immediate benefit of this type of software architecture is greater robustness for the entire device, since the software execution contained in one box can only affect the execution of the software contained in another box according to the defined communication rules. Therefore, a malfunction in one box cannot affect the function of the other boxes. The security benefits are obvious: the reliability of the software running in any box depends only on its own robustness and not on that of the other boxes. Furthermore, it is easy to prove that a given software box does not impact the certification of another software box.

The Common Approach

Cortex-M-based microcontrollers are widely used in the IoT world, due to their compact profile, low power consumption, and ease of use for designs running simple applications. They are usually preferred over Cortex-A-based microcontrollers, which typically target more elaborate designs that run complex applications based on large operating systems such as Linux or Android.

However, it is commonly accepted that the isolated software approach for security is not possible using Cortex-M-based microcontrollers (Figure 2) because the Cortex-M core does not have an MMU but instead has a Memory Protection Unit (MPU). Cortex-M-based micros are often considered very simple micros, where the embedded software is considered a single monolithic piece of code. The code is only as robust as its least robust portion. Consequently, it is widely believed that Cortex-M micros cannot fulfill all the necessary security requirements when the embedded application becomes large and complex.

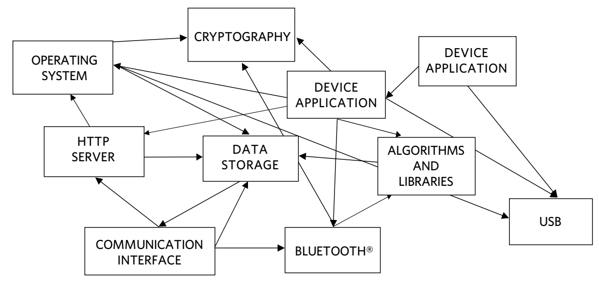

Figure 2. Typical Software Architecture on Cortex-M

DeepCover Security Framework

Our DeepCover® Security Framework isolation software makes it possible to reach a high level of security for large embedded applications running on Cortex-M-based micros (Figure 3). The solution guarantees strong isolation between the software "boxes" on the Cortex-M with MPU.

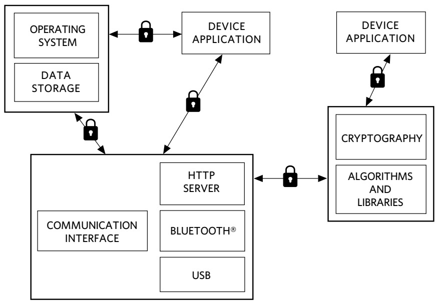

Figure 3. DeepCover Security Framework on Cortex-M

Security Framework Architecture

Our Security Framework consists of several components including a hypervisor that guarantees the integrity, authenticity, isolation, controlled communication, and privilege enforcement of boxes and other bricks (Figure 4). This brings the security approach to a system level:

- A secure bootloader provides the root of trust, verifying and controlling the origin, integrity, and authenticity of any software executed on the platform.
- A secure box provides a set of services for key management, cryptography, secure memory management, platform security management, and secure code update.
- Device drivers and libraries are bundled into the solution.
- A real-time operating system (RTOS) environment and a high level of service are included. By default, our solution relies on the Arm Mbed™ platform, the software framework proposed by Arm for the IoT market. This provides many turnkey software components like protocol stacks, RTOS, TLS, etc. However, it can also be integrated into other frameworks, e.g., using common RTOS such as FreeRTOS™.
- Complete, ready-to-build application templates are also provided to simplify final application development.

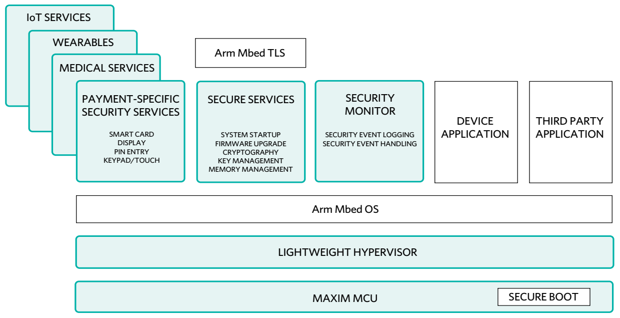

Figure 4. DeepCover Security Framework Architecture

PCI-PTS and IoT Applications

One of the applications of this solution is for payment terminal security certifications, PCI-PTS. A security evaluation report from a PCI-PTS-accredited lab concludes that the DeepCover Security Framework solution complies with the software security requirements emitted by PCI-PTS for software isolation.

Our DeepCover Security Framework solution is not only good for payment terminal manufacturers but also for IoT device manufacturers. Despite the lack of security standardization in the IoT world, meeting high security expectations as defined by PCI-PTS also brings a high level of trust to any IoT manufacturer.

Conclusion

The DeepCover Security Framework is a unique embedded isolated software solution which guarantees a high level of security for large embedded applications running on Cortex-M-based microcontrollers.

A full software development kit (SDK), based on well-known development tools such as GNU Make, GCC, GDB, and Eclipse is provided to simplify the complete application development.

The DeepCover Security Framework solution enables simplified development with a high level of security even on common platforms, while having very limited impact on platform performance and resources.

Learn more:

For more information on the DeepCover Security Framework, contact: secure.micro.support@maximintegrated.com

Visit our website for additional information on DeepCover Embedded Security Technology

Rev 0; November 2017

Visit the Maxim Support Center

Maxim Integrated
160 Rio Robles
San Jose, CA 95134 USA
408-601-1000
maximintegrated.com/deepcover

Maxim Integrated and the Maxim logo are registered trademarks of Maxim Integrated Corporation. All other trademarks are the property of their respective owners.