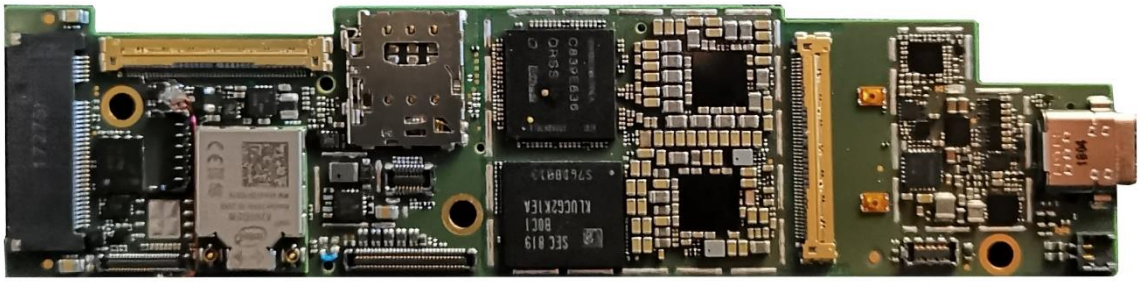
Whitepaper

This whitepaper describes the possibility to combine coreboot and LinuxBOOT to provide a secure and flexible solution to boot a modern Intel based system. This is just an example of a boot solution that can be created for a modern embedded system.



USING LINUXBOOT WITH SECURE COREBOOT

THE CHALLENGE

Our customer was looking for a Video Conferencing system with the following requirements:

- The system is deployed in high volume and on many locations. Therefore, the system should not require any manual setup or maintenance.
- It requires frequent software updates that should be applied without effort from the management side.
- It should boot as fast as possible.
- The Operating System image is available on the network and should be obtained reliably and securely especially when booting from remote servers on the internet.
- All of this should fit in an 8 MB flash device so there isn't much space in the device for complicated solutions.

The previous generation Video Conferencing system was based on a regular UEFI BIOS and was using PXE boot to load the Operating System image on each boot. This combination was taking too much time and was not secure and reliable.

USING LINUXBOOT WITH SECURE COREBOOT

Whitepaper

THE SOLUTION

To create the final result several steps were needed to achieve the desired result.

What makes this solution unique is the fact that LinuxBoot is used as the Operating System Loader. By using LinuxBoot this can be implemented in a very secure and efficient way as a proven network stack and well tested tools can be used. Because of this the implementation of the Operating System Loader can be done by configuring Linux, creating scripts and creating very limited custom software components. This Linux based loader uses the embedded eMMC drive as a cache instead of downloading the full OS image each time the system is started.

As the LinuxBoot will configure the required hardware devices by itself there is no need do this in coreboot. Now coreboot will only initialize the required components, enumerate the PCI bus and publish the required ACPI information. LinuxBoot which is used to load the final Operating System is integrated into the flash device eliminating the need to initialize and implement handlers for any mass storage device. By doing this the goals of reducing the boot time and providing sufficient space in the flash to fit the LinuxBoot image were achieved.

As the size of the flash device was 8 MB of which 2 MB are occupied by the Management Engine only 6 MB are available for the Linux kernel and the other components. By reducing the size of the coreboot components to less than 160 KB and carefully locating the Intel FSP and Microcode 5.5 MB of flash is available to fit the Linux kernel and filesystem.

For this solution two conflicting items are of the utmost importance, flexibility and security. It should be easy to update the flash to perform an update of LinuxBoot when needed. While on the other hand it should be absolutely sure that only the known LinuxBoot image will be booted and nothing else. This challenge was addressed using a custom designed verified boot implementation combined with measured boot.

The verified boot uses a manifest that contains SHA512 hashes of all components in flash. The manifest is created when the coreboot image is built. This manifest is signed during the build process and is contained in an area of the flash that can be updated. This manifest is used by the coreboot implementation to validate each software component before using it.

A small area of the flash (16 KB) is locked down and can't be updated. This area contains the code and public key necessary to verify the manifest. By using this implementation we can have a secure implementation of which only 16 KB is locked down.

To make sure the boot chain can be verified by the Operating System measured boot is implemented as well.

USING LINUXBOOT WITH SECURE COREBOOT

Whitepaper

USED HARDWARE AND SERVICES

This solution uses an Intel Braswell SoC based Com Express module combined with a custom carrier board.

The hardware is initialized using the Intel FSP combined with a coreboot and LinuxBoot solution created using Eltan's Hardware and Firmware Development services.

FUTURE GENERATION

A future generation of the product will be based on the Intel Apollo Lake SoC using the Intel FSP for Apollo Lake and the Intel Slim Bootloader. The new generation will provide upgraded video performance, improved camera interface, enhanced security and better QoS behavior.

Eltan's Hardware and Firmware Development services will be used to implement the solution to this hardware and provide the security and flexibility required.

ELTAN B.V.

Eltan B.V.
Schijndel, The Netherlands
tel: +31-73-594 46 60

www.eltan.com
info@eltan.com