x86 Boot Chain and Secure Boot Training: UEFI, UKI

UEFI, boot loaders and unified kernel images: the boot chain of an x86 product, from the firmware to an updatable, measured and signed Linux.

2 days	Intermediate to advanced	On site or remote	English or French
Duration	Level	Format	Language

Overview

This course covers the boot chain of an embedded Linux product built on x86, and how to secure it. The UEFI firmware comes from the board vendor and is not ported: the work is in the EFI System Partition and the EFI variables, the choice between GRUB, systemd-boot or a unified kernel image started by the firmware, the Secure Boot keys and the TPM, and the updates that must keep all of it bootable.

The course is drawn from our work on industrial x86 PCs. For Arm platforms, the Arm boot chain and secure boot course covers the ROM code, TF-A, OP-TEE and U-Boot.

What You Will Learn

- Explain how UEFI finds and starts Linux, and what the EFI variables control
- Choose between GRUB, systemd-boot and a unified kernel image booted directly
- Provide ACPI overrides and CPU microcode early in the boot
- Generate, enroll and protect the UEFI Secure Boot keys, and use shim and MOK when needed
- Bind secrets to the boot state with the TPM and PCR 11
- Update the boot chain safely with A/B slots and BootNext

Who should attend

Engineers who build embedded Linux products on x86 boards, industrial PCs or Intel development kits, and need to boot and secure them.

Prerequisites

Good knowledge of embedded Linux. Basic knowledge of the Yocto Project helps for the integration parts.

Program

The first day covers how an x86 board boots Linux, the second how to secure and update that chain.

Day 1

- Firmware in the boot chain: legacy BIOS and UEFI
- The EFI System Partition, EFI variables and efivarfs
- Booting Linux: the EFI stub, systemd-boot, GRUB, and unified kernel images with or without systemd-stub
- ACPI tables and their overrides, CPU microcode
- Image formats and the integration in a Yocto build

Day 2

- UEFI Secure Boot: the PK, KEK and db key hierarchy, key generation and enrollment
- shim, machine owner keys and SBAT revocation
- Unified kernel images and the TPM: PCR 11 and sealed secrets
- A/B updates with RAUC and BootNext
- Pitfalls met in production: a broad CA in db, keys on the build server, revocation never planned

Labs and Hardware

Labs run on QEMU machines and on the boards of our lab: Arm 32-bit: STM32MP157C-DK2, BeagleBone Black; Arm 64-bit: Raspberry Pi 4 and 5, NXP i.MX8M Mini, Khadas VIM3; x86: QEMU x86, Generic x86_64 machines, Intel development kits. Other boards, or the hardware of the customer, can be used on request.

Sessions take place on site, in France, Belgium, Switzerland, Germany and beyond, with the hardware provided by LABCSMART, or remotely. Remote attendees work on our build server, each with their own SSH account and workspace, and run the labs on QEMU machines and on real boards. The exercises are drawn from our own client projects. Courses are given in English or French.

Who Teaches This Course

The course is designed and given by the founder of LABCSMART, an embedded Linux and Linux kernel engineer with more than eleven years of industrial experience. He contributes upstream to the Linux kernel for Renesas RZ SoCs, in the PCIe, audio, Ethernet, thermal, DMA and clock subsystems, and wrote two books published by Packt, *Linux Device Driver Development, Second Edition* (2022) and *Mastering Linux Device Driver Development* (2021), presented on our publications page.

Plan a session for your team. Every course is made of independent modules. A team can drop the modules it does not need and add others in their place, and we also build courses on demand around a given SoC, board or project. Contact: contact@labcsmart.com, https://labcsmart.com/company/contact/. Course page: https://labcsmart.com/training/x86-boot-chain-secure-boot/