Formation chaîne de boot Arm et secure boot : U-Boot, TF-A

Du code ROM au noyau : la chaîne de boot d'une carte Arm, intégrée, portée et sécurisée étage par étage.

4 jours	Intermédiaire à avancé	Sur site ou à distance	Français ou anglais
Durée	Niveau	Format	Langue (supports en anglais)

Présentation

Cette formation suit la chaîne de boot d'une carte Arm, du code ROM jusqu'au noyau, l'intègre sur une nouvelle carte et la sécurise de bout en bout. Elle couvre les premiers chargeurs, le monde sécurisé avec Trusted Firmware-A et OP-TEE, le bootloader qui lance Linux, les formats d'image qui les transportent, le portage sur un matériel spécifique, et une journée entière sur le secure boot : authentification par le code ROM, images signées et bootloader verrouillé.

Les exemples utilisent U-Boot, TF-A et OP-TEE sur des SoC i.MX8M Mini, STM32MP15 et TI AM64x, et les notions s'appliquent aussi aux autres bootloaders. Pour les plateformes x86, la formation chaîne de boot x86 et secure boot couvre UEFI, les boot loaders et les unified kernel images.

Objectifs

À la fin de la formation, les participants savent :

- expliquer qui charge quoi, depuis où et à quel niveau d'exception sur un SoC Arm 64 bits ;
- intégrer TF-A, OP-TEE et U-Boot dans une chaîne de boot fonctionnelle ;
- assembler une image de boot complète aux offsets attendus par le code ROM ;
- porter la chaîne de boot sur une carte spécifique ;
- sécuriser la chaîne : authentification par la ROM, images FIT signées, trusted boot FIP et bootloader verrouillé ;
- déboguer une carte qui ne démarre pas.

Public

Ingénieurs BSP qui démarrent de nouvelles cartes, et développeurs qui maintiennent le firmware de boot d'un produit ou doivent le sécuriser.

Prérequis

Bonne connaissance de Linux embarqué et du C. Des notions sur les processeurs Arm sont un plus.

Programme

Les trois premiers jours suivent la chaîne, du code ROM jusqu'à une carte portée, le quatrième la sécurise.

Jour 1

- La chaîne de boot d'un système Linux embarqué : le code ROM et les étages successifs
- Le monde sécurisé Arm : TrustZone, Trusted Firmware-A et OP-TEE
- Récupérer et compiler U-Boot
- Le shell du bootloader : inspecter la carte, le stockage, le réseau, les scripts

Jour 2

- L'environnement : variables, stockage et accès depuis Linux
- Démarrer Linux à la main : ce qu'attend le noyau et le device tree qu'il reçoit
- La séquence d'initialisation du bootloader et sa relocation
- La SPL : rôle, contraintes et recherche de l'étage suivant

Jour 3

- Le device tree dans le bootloader et les fixups appliqués au device tree du noyau
- Le driver model
- Assembler les images de boot avec binman
- Ajouter le support d'une nouvelle carte, des premières instructions jusqu'au portage complet
- Standard boot et boot UEFI

Jour 4

- Les principes du secure boot et l'authentification du premier chargeur par le code ROM, avec les clés dans les fuses
- Les images FIT et le verified boot avec des configurations signées
- Les images FIP et le trusted board boot de TF-A
- Verrouiller le bootloader : environnement, console et commandes de boot
- Le secure boot dans un build Yocto, et ses pièges
- Déboguer une carte qui ne démarre pas

Travaux pratiques et matériel

Les travaux pratiques se font sur des machines QEMU et sur les cartes de notre laboratoire : Arm 32 bits : STM32MP157C-DK2, BeagleBone Black ; Arm 64 bits : Raspberry Pi 4 et 5, NXP i.MX8M Mini, Khadas VIM3 ; x86 : QEMU x86, Machines x86_64 génériques, Kits de développement Intel. D'autres cartes, ou le matériel du client, peuvent être utilisés sur demande.

Les sessions ont lieu sur site, en France, en Belgique, en Suisse, en Allemagne et au-delà, avec le matériel fourni par LABCSMART, ou à distance. À distance, chaque participant dispose de son propre compte SSH et de son espace de travail sur notre serveur de build, et réalise les travaux pratiques sur des machines QEMU et sur de vraies cartes. Les exercices sont tirés de nos projets clients. Les formations sont données en français ou en anglais ; les supports de cours, eux, sont en anglais.

Modules optionnels

Ces modules complètent la formation pour les équipes qui développent le bootloader lui-même.

- Écrire des commandes, logs et règles de codage (une demi-journée)
- Déboguer avec une sonde, le framework de test et la contribution upstream (1 jour)

Qui anime cette formation

La formation est conçue et animée par le fondateur de LABCSMART, ingénieur Linux embarqué et noyau Linux avec plus de onze ans d'expérience industrielle. Il contribue au noyau Linux en amont pour les SoC Renesas RZ, dans les sous-systèmes PCIe, audio, Ethernet, thermique, DMA et horloges, et a écrit deux livres publiés chez Packt, *Linux Device Driver Development, Second Edition* (2022) et *Mastering Linux Device Driver Development* (2021), présentés sur notre page de publications.

Organiser une session pour votre équipe. Chaque formation est composée de modules indépendants. Une équipe peut retirer les modules dont elle n'a pas besoin et en ajouter d'autres à la place, et nous construisons aussi des formations sur mesure autour d'un SoC, d'une carte ou d'un projet. Contact : contact@labcsmart.com, https://labcsmart.com/fr/contact/. Page de la formation : https://labcsmart.com/fr/formations/formation-chaine-de-boot-arm-secure-boot/