




## Formation sécurité des systèmes Linux embarqués

Durée de la formation \_\_\_\_\_

 4 demi-journées – 16 h

Langue \_\_\_\_\_

Transparents    Anglais

Présentation    Français  
                          Anglais


Formateur \_\_\_\_\_

Un des ingénieurs suivants

- Mathieu Dubois-Briand
- Olivier Benjamin

Contact \_\_\_\_\_

 training@bootlin.com

 +33 4 84 25 80 96



### Public visé

Entreprises et ingénieurs qui conçoivent, développent et produisent des systèmes Linux embarqués ayant des exigences en matière de cybersécurité.

### Objectifs opérationnels

- Être capable de comprendre les mécanismes de sécurité et d'isolation des systèmes Linux embarqués modernes : mémoire non exécutable, randomisation de l'espace d'adressage, niveaux de privilèges, contrôle d'accès obligatoire et discrétionnaire.
- Être capable de concevoir et de mettre en œuvre une chaîne de démarrage sécurisée, du chargeur de démarrage jusqu'à l'espace utilisateur, y compris dm-verity.
- Être capable de raisonner sur la cryptographie et de mettre en œuvre un stockage sécurisé des clés, à l'aide d'un HSM matériel ou logiciel.
- Être capable d'exploiter la technologie ARM TrustZone (monde sécurisé, TF-A, OP-TEE) afin d'isoler les opérations sensibles et de développer des Trusted Applications.
- Être capable de mettre en œuvre des mécanismes de sécurité en espace utilisateur : capacités Linux, namespaces, cgroups, SECCOMP, SELinux et fonctionnalités de durcissement de systemd.
- Être capable de configurer le chiffrement du système de fichiers à l'aide de dm-crypt et de l'intégrer à une gestion sécurisée des clés via PKCS#11.
- Être capable de gérer les vulnérabilités logicielles à l'aide de bases de données CVE, de générer et d'analyser des Software Bills of Materials (SBoM), et de comprendre les exigences de conformité, notamment le Cyber Resilience Act.
- Être capable de concevoir et de déployer des mécanismes de mise à jour sécurisés à l'aide de schémas de partitionnement A/B et d'outils tels que RAUC ou SWUpdate.
- Être capable de comprendre les concepts de measured boot et de mettre en œuvre la vérification de l'intégrité de la plateforme à l'aide de TPM et de IMA/EVM.

### Prérequis

- **Connaissance et pratique des commandes UNIX ou GNU/Linux** : les participants doivent être à l'aise avec l'utilisation de la ligne de commande Linux. Les participants manquant d'expérience sur ce sujet doivent se former par eux-mêmes, par exemple en utilisant nos supports de formation.
- **Expérience minimale en développement Linux embarqué** : les participants doivent avoir une compréhension minimale de l'architecture d'un système Linux embarqué : rôle du noyau Linux par rapport à l'espace utilisateur, développement d'applications espace utilisateur en C. Suivre la formation Linux embarqué de Bootlin permet de remplir ce pré-requis.
- **Niveau minimal requis en anglais** : **B1**, d'après le *Common European Framework of References for Languages*, pour nos sessions animées en anglais. Voir la grille CEFR pour une auto-évaluation.

### Méthodes pédagogiques

- Présentations animées par le formateur, par visioconférence. Les participants peuvent poser des questions à tout instant.
- Démonstrations pratiques réalisées par le formateur, basés sur les travaux pratiques de la formation, par vidéo-conférence. Les participants peuvent poser des questions à tout instant. Optionnellement, les participants qui ont accès aux accessoires matériels de la formation peuvent reproduire par eux-même les travaux pratiques.
- Messagerie instantanée pour questions entre les sessions (réponse sous 24h, hors week-end et jours fériés)
- Version électronique des supports de présentation, des instructions et des données de travaux pratiques. Les supports sont librement disponibles [ici](#).

### Modalités d'évaluation

Seuls les participants qui auront assisté à l'intégralité des journées de formation, et qui auront obtenu plus de 50% de réponses correctes à l'évaluation finale recevront une attestation individuelle de formation de la part de Bootlin.

### Handicap

Les participants en situation de handicap qui ont des besoins spécifiques sont invités à nous contacter à l'adresse [training@bootlin.com](mailto:training@bootlin.com) afin de discuter des adaptations nécessaires à la formation.

## Équipement nécessaire

Équipement obligatoire :

- Ordinateur avec le système d'exploitation de votre choix, équipé du navigateur Google Chrome ou Chromium pour la conférence vidéo.
- Une webcam et un micro (de préférence un casque avec micro)
- Une connexion à Internet à haut débit

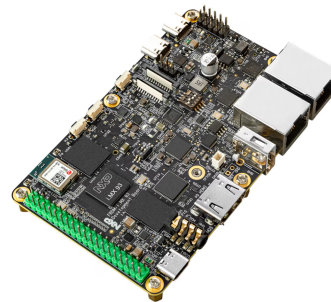
Optionnellement, si les participants souhaitant pouvoir reproduire par eux-mêmes les travaux pratiques, ils doivent acheter séparément la carte de développement et les accessoires associés, et devront disposer d'un PC avec une installation native d'Ubuntu Linux 24.04.

## Plateforme matérielle pour les travaux pratiques

### NXP i.MX93 FRDM

Carte de développement **NXP FRDM-IMX93**

- Processeur NXP i.MX93 (Dual Cortex-A55 + Cortex-M33)
- 2 GB LPDDR4X, 32 GB eMMC
- Double Ethernet Gigabit
- USB 2.0 Type-C + USB Type-A
- Interface CAN
- Slot microSD, EEPROM
- Wi-Fi 6 + Bluetooth 5.4 + 802.15.4 (MAYA-W276)
- Sortie HDMI (via LVDS), MIPI DSI et CSI
- Audio jack (MQS), boutons and LEDs
- debug via SWD et UART



## Demi-journée 1

Cours	Concepts fondamentaux	<ul style="list-style-type: none"> <li>Modélisation des menaces</li> <li>Bases de la cryptographie : clé publique / clé privée</li> <li>Comprendre un handshake TLS du début à la fin</li> <li>Comprendre les infrastructures à clés publiques (PKI)</li> </ul>
Démo	Configuration d'une PKI simple	<ul style="list-style-type: none"> <li>Utilisation d'OpenSSL pour générer une hiérarchie de clés</li> <li>Expérimentation de la révocation de clés</li> <li>Comparaison des performances du chiffrement par clé publique/clé privée</li> </ul>
Cours	Limites de sécurité assistées par le matériel	<ul style="list-style-type: none"> <li>Concepts généraux de sécurité : séparation noyau/utilisateur, CPL, protections mémoire</li> <li>Contre-mesures courantes : ASLR, NX/DEP, SMAP/SMEP, RELRO</li> <li>Fonctionnalités ARM : niveaux d'exception</li> <li>Fonctionnalités ARM : <i>secure world</i>, TF-A, OP-TEE</li> <li>Compréhension d'un flux de démarrage complet ARMv8, y compris le monde sécurisé</li> <li>Compréhension des Trusted Applications</li> </ul>
Démo	Exploration du <i>secure world</i>	<ul style="list-style-type: none"> <li>Compilation de logiciel pour le <i>secure world</i> (TF-A, OP-TEE, TA)</li> <li>Ajout de journaux pour observer les transitions de niveaux d'exception</li> <li>Ajout de journaux pour observer les transitions entre mondes</li> <li>Écriture d'une petite Trusted Application dans OP-TEE</li> <li>Interaction avec notre Trusted Application depuis l'espace utilisateur</li> </ul>

## Demi-journée 2

Cours	Secure boot : première partie	<ul style="list-style-type: none"> <li>Compréhension du concept de chaîne de démarrage sécurisée</li> <li>Exemples de mise en œuvre du secure boot : x86 et SoC ARM</li> <li>Secure boot : modèle de menace</li> <li>Être capable de concevoir et de mettre en œuvre une chaîne de démarrage sécurisée</li> </ul>
Démo	Secure boot sur i.MX93	<ul style="list-style-type: none"> <li>Compilation de tous les composants logiciels pour le secure boot (AHAB) sur i.MX93</li> <li>Création d'un conteneur AHAB signé à l'aide de SPSDK</li> <li>Programmation des fusibles sur l'i.MX93 pour activer le secure boot (AHAB)</li> <li>Vérification que le secure boot (AHAB) est correctement activé</li> </ul>
Cours		<ul style="list-style-type: none"> <li>Analyse approfondie des composants d'U-Boot : TPL/SPL, U-Boot principal, images FIT</li> <li>Description matérielle : Device Tree</li> <li>Activation de la vérification des signatures dans le chargeur de démarrage U-Boot</li> <li>Vérification du RootFS : <i>dm-verity</i> et implications pour la conception du système</li> </ul>
Démo	Secure boot sur i.MX93 : étape du bootloader	<ul style="list-style-type: none"> <li>Intégration de la clé publique dans le DTB</li> <li>Ajout d'une signature à l'image FIT du noyau</li> <li>Configuration de U-Boot pour imposer la vérification des signatures</li> <li>(optionnel) Configuration du SPL pour imposer également la vérification des signatures</li> </ul>

## Demi-journée 3

Cours	Protection des données	<ul style="list-style-type: none"> <li>▪ Chiffrement du système de fichiers, présentation de dm-crypt/cryptsetup</li> <li>▪ Gestion des clés : module matériel de sécurité (HSM)</li> <li>▪ Utilisation sécurisée des clés : introduction à PKCS#11</li> <li>▪ Utilisation sécurisée des clés : trousseau de clés du noyau et clés de confiance</li> </ul>
Démo	Gestion des clés	<ul style="list-style-type: none"> <li>▪ Provisionnement des clés dans l'ELE du i.MX93</li> <li>▪ Utilisation d'OP-TEE comme HSM logiciel</li> <li>▪ (optionnel) Chiffrement du système de fichiers à l'aide de la clé ELE</li> <li>▪ (optionnel) Signature des images FIT de U-Boot avec intégration d'un HSM via PKCS#11</li> </ul>
Cours	Mécanismes de sécurité en user-space	<ul style="list-style-type: none"> <li>▪ Paradigmes de contrôle d'accès : MAC/DAC</li> <li>▪ Linux <i>capabilities</i> : utilisation et exemples</li> <li>▪ Isolation des processus : namespaces, cgroups, SECCOMP</li> <li>▪ Modules de sécurité Linux : SELinux, AppArmor</li> <li>▪ <i>Hardening</i> des applications via <i>systemd</i></li> </ul>
Démo	Restriction des applications user-space	<ul style="list-style-type: none"> <li>▪ Empêcher une application simple d'exécuter du shellcode à l'aide de SECCOMP</li> <li>▪ Manipulation des contextes SELinux</li> <li>▪ Utilisation de <i>systemd</i> pour restreindre l'accès d'un démon aux ressources</li> </ul>

## Demi-journée 4

Cours	Maintenance, réglementations et conformité	<ul style="list-style-type: none"> <li>▪ Introduction aux cadres de gestion des vulnérabilités : CVE, CWE, CVSS</li> <li>▪ Présentation du Cyber Resilience Act (CRA)</li> <li>▪ Définition d'une stratégie de mise à jour, basée sur les cycles de publication et de maintenance des projets open source importants</li> <li>▪ Génération d'un Software Bill of Materials</li> <li>▪ Analyse d'un Software Bill of Materials</li> </ul>
Démo	Prise en main des SBoM et des CVE	<ul style="list-style-type: none"> <li>▪ Génération d'un SBoM pour une distribution Yocto</li> <li>▪ Analyse du SBoM généré à l'aide de <i>sbom-cve-check</i></li> <li>▪ Correction d'une CVE présente dans le SBoM</li> <li>▪ Analyse différentielle</li> </ul>
Cours	Mises à jour sécurisées	<ul style="list-style-type: none"> <li>▪ Mises à jour A/B</li> <li>▪ Présentation de SWUpdate</li> <li>▪ Présentation de RAUC</li> <li>▪ Intégration au chargeur de démarrage</li> <li>▪ Conséquences sur la chaîne de démarrage sécurisée</li> </ul>
Démo	Mise en oeuvre de RAUC pour les mises à jour sécurisées	<ul style="list-style-type: none"> <li>▪ Configuration de RAUC sur la cible</li> <li>▪ Création, déploiement et installation d'un bundle RAUC</li> <li>▪ Vérification de la signature du bundle à l'aide de PKCS#11</li> <li>▪ (optionnel) Chiffrement du bundle RAUC</li> <li>▪ (bonus) Intégration d'un HSM</li> </ul>