SEC8 - Construction d'une plate-forme Linux embarqué sécurisée

Objectifs

- Mettre en œuvre le secure boot
- Vérifier l'authenticité des composants du système avant leur chargement et leur exécution.
- Garantir l'authenticité et l'intégrité du bootloader et du noyau
- Mettre en œuvre le Trusted Boot
- Fournir un environnement sécurisé au firmware du secure monitor
- Exécuter OP-TEE dans un environnement sécurisé qui fonctionne aux côtés du système d'exploitation principal

Prérequis

- Connaissance du langage C (voir par exemple notre formation L2)
- Connaissance de la construction d'un Linux embarqué (voir par exemple notre formation D1)
- Le cours SEC9 Sécurité avancée du Linux embarqué peut également vous intéresser
- Le cours SEC1 Développement sécurisé pour systèmes embarqués peut également vous intéresser
- Le cours SEC2 Sécurité avancée des systèmes embarqués peut également vous intéresser

Matériel

- Supports de cours et exercices logiciels
- Un PC Linux pour deux stagiaires
- Une plate-forme cible pour deux stagiaires

Durée

- Total : 2 jours
- De 40% à 50% du temps de formation est consacré aux activités pratiques

Environnement du cours

- Cours théorique
  - Support imprimé et PDF (en anglais).
  - Assistance du formateur tout au long de la formation.
- Chaque session débute par un point avec les stagiaires.

Audience visée

- Tout ingénieur ou technicien en systèmes embarqués possédant les prérequis ci-dessus.

Plan du cours

Premier jour

Présentation de Linux

- Histoire de Linux
- Architecture et modularité de Linux
- Composants d'un système Linux
- Les différentes licences utilisées par Linux (GPL, LGPL, etc)

Chaîne de démarrage

- Démarrage bas niveau
  - Démarrage sur NOR
  - Démarrage sur NAND
  - Démarrage sur SD/MMC/eMMC
  - Démarrage multi-étages
  - Pourquoi une chaîne de démarrage de confiance est-elle nécessaire
- Enjeux de sécurité
  - Confidentialité et intégrité
  - Prévention des altérations
  - Conformité et certification

Secure Boot

- Le concept de secure boot
  - La chaîne de confiance
  - Le processus complet de secure boot
- Gestion des clés
  - Introduction à la gestion des clés
  - Algorithmes cryptographiques et types de clés
  - Options de stockage des clés : matérielles et logicielles
  - Processus de gestion des clés : génération et révocation
  - Vue d'ensemble des fonctions matérielles des plates-formes ARM
  - Secure Monitor
  - Secure World
  - Trusted Execution Environment
  - Secure boot sur RISCV et X86_64
  - Accélérateurs cryptographiques
- Solutions logicielles
  - Open source
  - Propriétaires

Bootloaders de premier et second étage

- U-Boot
  - Possibilités et fonctionnalités
  - Configuration, personnalisation et compilation
  - U-Boot SPL comme First-Stage Boot Loader (SSBL)
  - Rôle d'u-boot dans la chaîne de démarrage de confiance
  - Comment U-Boot vérifie l'authenticité des images qu'il charge
  - Options de configuration pour sécuriser le processus de démarrage
  - Interaction avec le secure world et le Trusted Execution Environment

    - Signature d'U-boot
- Arm Trusted Firmware (ATF)
    - Présentation et fonctionnalités
    - Séquence de démarrage de l'ATF
    - Services
    - Construction et déploiement
- Autres composants spécifiques à la plate-forme

Image Linux sécurisée

- Introduction au noyau Linux
    - Code source
    - Configuration
    - Compilation
- Image FIT (Flattened Image Tree)
    - Qu'est-ce que le FIT et pourquoi l'utilise-t-on
    - Avantages de l'utilisation d'une image FIT
    - Configuration
    - Construction d'une image FIT sécurisée
- Configuration du noyau pour une plate-forme Linux sécurisée
    - Options de configuration du secure boot dans le noyau Linux
    - Vue d'ensemble de la configuration du contrôle d'accès

Deuxième jour

Considérations de sécurité lors de la création d'un système de fichiers racine

- Conseils pour durcir et sécuriser un rootfs
    - Réduire le rootfs au minimum
    - Authentification forte
    - Maintenir les logiciels à jour
    - Utiliser un initramfs
- Système de fichiers racine en lecture seule
    - Introduction au système de fichiers racine en lecture seule
    - Objectif et bénéfices
    - Vue d'ensemble des différentes solutions disponibles
    - SquashFS
    - CramFS : faible empreinte mémoire
    - Système de fichiers racine en lecture seule basé sur OverlayFS
    - Système de fichiers racine en lecture seule basé sur UnionFS
- Critères de choix d'une solution de système de fichiers racine en lecture seule
    - Évaluation selon le cas d'usage, la sécurité, les performances et la compatibilité
- Chiffrement des images de mise à jour
    - Mettre à jour de façon sécurisée une plate-forme Linux avec Mender

Open Portable Trusted Execution Environment (OP-TEE)

- Introduction à OP-TEE
- Fonctionnalités clés
- Prérequis matériels, logiciels et firmware
- Architecture d'OP-TEE
    - Composants, modules et canaux de communication
- Cas d'usage
    - Stockage sécurisé
    - Communication sécurisée
    - Exécution sécurisée d'applications
- Construction et déploiement d'OP-TEE

    - Mise en place de l'environnement
    - Configuration d'OP-TEE
    - Compilation d'OP-TEE
- Comparaison avec les autres solutions de TEE
- Trusted Applications (TA) sur OP-TEE
    - Le rôle d'une TA dans un système sécurisé
    - Écriture d'une Trusted Application
    - Chargement et exécution d'une Trusted Application dans le runtime OP-TEE
    - Débogage et test des Trusted Applications
    - Communication entre les Trusted Applications et les applications du monde normal
    - Bonnes pratiques pour créer des Trusted Applications sécurisées