MSE | MASTER OF SCIENCE IN ENGINEERING

Secure Embeeded Systems

Laboratoire I

U-Boot & Linux Kernel

Savy Cyrille - Mueller Michael

2015

HES-SO//Master
Orientation TIC

Professeur	: Schuler Jean-Roland
Branche	: SES
Salle de laboratoire	: 5
Dernière mise à jour	: 7 avril 2015

Table des matières

Chapitre 1

Introduction

La sécurité dans les systèmes embarquées, qui sont présent de plus en plus souvent dans un nombre de domaine grandissant, est primordiale pour éviter les abus de personnes malintentionnées (voir éviter les erreurs humaines possibles...).

Le but de ce laboratoire était de découvrir l'environnement U-Boot sur la plateforme "Odroid" basée sur un processeur "Samsung", l'Exynos5422 ! C'est une plateforme beaucoup plus performante que celles utilisée précédement au cours de CSEL, qui elle est basée sur un processeur Freescale AFP27.

Dans ce laboratoire, nous avons pu appliquer certaines méthodes de software hardening en changeant quelques commandes de compilation, notamment pour ôter les symboles de "debug".

Chapitre 2

Bootloader U-Boot

2.1 MMC partitionning

2.1.1 Automated MMC partitionning

2.2 U-Boot hardening

2.2.1 GCC compilation command line

2.2.1.1 Debug symbol stripping

2.2.1.2 Using canaries

Chapitre 3

Conclusion

Ce laboratoire nous a permis de mettre en pratique la théorie vue en cours ainsi que de rafraichir nos connaissances précédentes sur Linux embarqué. Les systèmes embarqué étant "limités" (cette affirmation est aujourd'hui à moitié vraie... Notre plateforme est tout de même dotée d'un octo-coeur ARM et de beaucoup de mémoire vive. Mais ce n'est pas forcément le cas de tous les systèmes sur le marché) en performances nécessitent un traitement particulier quand à leur sécurité.

On peut facilement prendre ces systèmes et en extraire les données. Une partie de la sécurité peut être crée à la conception hardware du système (protection des ports JTAG, etc...), mais le reste doit être géré au niveau du software.

Nous avons vu quelques techniques de protections en modifiant les variables de compilation. L'utilisation de la cryptographie devrait être mieux appliquée au niveau du bootloader (notre version ne la support pas, mais elle existe dans les versions supérieure). Contrôler le noyau chargé en mémoire par un hash et/ou une signature (avec ou sans certificats) serait adéquat.
La protection du système de fichiers fait partie de la seconde partie du laboratoire.