Secure boot loader et attaques par faute

Contexte

Les attaques en faute ont longtemps été vues comme une menace spécifique aux composants sécurisés de type carte à puce pour lesquels un processus de certification, obligatoire avant mise sur le marché, inclut une recherche de vulnérabilité très poussée (pour avoir le niveau AVA_VAN5). Toutefois, les attaques en faute sont aujourd'hui une menace pour un grand nombre de cibles : elles sont réalisables de manière matérielle à moindre coût matériel et humain et de plus, elles peuvent être produites de manière 100% logicielle.

Ces attaques dont l'objectif est de perturber l'exécution afin de prendre la main sur le système ou provoquer une déviation dans les calculs ou le flot de contrôle sont puissantes, elles permettent notamment de retrouver des informations secrètes (clés cryptographiques) ou de contourner des mécanismes d'authentification.

Elles sont particulièrement dangereuses lors du démarrage d'un système, car elles permettent de prendre le contrôle du système avant tout déploiement de mécanisme de sécurité à différents niveaux de la pile logicielle.

Ajouter des protections contre ces attaques est difficile : il faut cibler les éléments sensibles, ajouter du code de protection et garantir son efficacité. De plus, la quantité de protection à ajouter dépend du système final, de son utilisation ainsi que des données manipulées sensibles à protéger. Lors de la conception des systèmes, les attaques en faute ne sont pas toujours considérées par manque d'expertise, de temps ou de ressources (humaines et matérielles). Aujourd'hui, seuls les recherches de vulnérabilités avancées, menées dans les centres de certification, attestent d'une forte résistance face à ces attaques ; il peut néanmoins être choisi de déployer des protections typiques et arriver à un niveau de résistance aussi élevé même sans demander la certification correspondante. Il est aussi important de pouvoir déployer les protections nécessaires pour obtenir des niveaux avancés de certifications. Pour cela, il apparaît essentiel de savoir déployer des protections contre une large gamme de fautes possibles tout en laissant le choix au concepteur final des protections nécessaires et adaptées à un produit.

Pour toutes ces raisons, des moyens permettant de faciliter le choix et l'intégration de protections contre ces attaques en faute dans le code de démarrage pour atteindre un niveau de sécurité souhaité par le concepteur sont à proposer.

Objectif

L'objectif de ce stage est d'étudier l'efficacité de protections existantes dans un *secure boot loader* open source (MCU boot) et d'analyser leur pertinence vis à vis de certains critères de certifications. L'analyse de l'efficacité se fera par simulation d'injection de faute réalisée à partir d'outils existants qu'il conviendra d'adapter le cas échant. De plus, le stagiaire pourra proposer des améliorations de l'implémentation des protections de MCU boot pour les rendre plus robustes, tant aux attaques qu'aux optimisations de code.

Déroulement du stage (indicatif car à adapter au stagiaire recruté)

- Prise en main de MCU Boot, de la plateforme de simulation matérielle incluant la possibilité de simuler des fautes et d'une plateforme matérielle cible
- Bibliographie sur l'injection de faute, les protections logicielles, la certification
- Revue de code et définition de campagnes automatisée d'analyse de robustesse de MCU Boot (protections incluses, compilateur et optimisation)

- Potentielle campagne d'injection de faute réelle en collaboration avec l'ANSSI
- Proposition d'amélioration
- Rédaction d'un document décrivant les protections, leur localisation ainsi que leurs effets attendus, leur efficacité selon les campagnes réalisées et leur nécessité vis à vis des attentes de la certification.

Encadrement, durée et lieu du stage

- Le stagiaire sera essentiellement encadré par A. de Grandmaison (Arm). Toutefois le stage s'inscrit dans une collaboration, et il y aura deux co-encadrants extérieurs : K. Heydemann de Sorbonne Université et G. Bouffard de l'ANSSI.
- Ce sujet vise un stage de fin d'étude (Master 2, école d'ingénieurs) d'une durée d'environ 6 mois qui doit avoir lieu en 2021.
- Le stage se déroulera dans les bureaux de Arm à la Défense avec des réunions régulières d'avancement sur le campus de Jussieu à Paris ; le stagiaire aura aussi la possibilité d'aller dans les locaux de l'ANSSI pour mener des expérimentations sur banc encadrées et gérées par G. Bouffard.
  Si les conditions sanitaires et les mesures en vigueur l'exigent, le stage pourra avoir lieu en télétravail.

Profil du candidat recherché / compétences requises

- Architecture des CPU, des systèmes embarques et leur pile logicielle
- Développements logiciels (C, C++, Python) et goût pour l'intégration logicielle
- Compilation et optimisation, sécurité (connaissances bienvenues mais non requises)
- Esprit d'analyse et suivi d'une démarche scientifique
- Capacité à interagir avec différents encadrants et différentes équipes

Contact

Arnaud de Grandmaison : arnaud.degrandmaison@arm.com
Karine Heydemann : karine.heydemann@lip6.fr)
Guillaume Bouffard : guillaume.bouffardd@anssi.fr)

Références et bibliographie

- **Fault injection**
  B. Yuce, P. Schaumont, and M. Witteman. *Fault attacks on secure embedded software: Threats, design, and evaluation*. Journal of Hardware and Systems Security, 2(2):111–130, 2018.
- **Software protections**
  Marc Whiteman. *Secure Application Programming in the presence of Side Channel Attacks.* White paper by Riscure
- **Attacks and defense of secure boot**
  https://www.riscure.com/publication/secure-boot-under-attack-simulation-to-enhance-fault-attacks-defenses/
- **MCU Boot**
  https://github.com/mcu-tools/mcuboot