Modélisation et caractérisation d'attaques par faute exploitant l'architecture mémoire

2023-2024

Laboratoire VERIMAG (Grenoble)
Équipe PACS (Preuve et Analyse de Code pour la Sécurité)
Encadrants: Bruno Ferres & Pierre Corbineau
bruno.ferres@univ-grenoble-alpes.fr
pierre.corbineau@univ-grenoble-alpes.fr

Mots-clef: Sécurité matérielle; micro-architecture; simulation; modèles de faute

Contexte

Les attaques en fautes (par faisceau laser, injection électromagnétique, ou même purement logicielle) constituent une menace sévère contre tous les systèmes embarqués [1, 2]. Pour s'en protéger ou évaluer la résistance des composants, des simulateurs de fautes sont primordiaux pour cibler précisément les attaques, ou vérifier exhaustivement qu'aucun chemin d'attaque n'a été oublié [3].

Quelques simulateurs de fautes pour software existent aujourd'hui, dont CELTIC développé au CEA-LETI [4]. Ils sont encore souvent limités à des cas d'usage très simples, ne permettant pas leur utilisation courante pour les évaluations sécuritaires. Par exemple, la présence de périphériques, d'interruptions, de zone sécurisés dans les composants embarqués — en résumé, les détails micro-architecturaux des systèmes — sont rarement pris en compte.

Les modèles de faute actuellement étudiés s'appliquent en général à une description fonctionnelle de l'exécution des programmes (au niveau ISA – *Instruction-Set Architecture*). Pourtant un nombre croissant d'attaques laisse apparaître des caractéristiques plus bas niveau de l'architecture des processeurs ciblés (hiérarchie mémoire, tampons, pipelines).

Objectifs du stage

Plusieurs questions de recherche se posent alors :

- Modélisation : Comment peut-on simuler efficacement les effets de fautes affectant la micro-architecture ?
- Caractérisation : Comment peut-on différencier expérimentalement ces fautes d'autres classes de fautes plus communes ?
- Protection : Quelles contre-mesures (notamment logicielles) peut-on envisager ? Comment durcir le code exécuté ?

Ce sujet s'appuiera sur des travaux préliminaires menés dans l'équipe et pourra être décliné de différentes façons, suivant les intérêts du candidat/de la candidate.
Suivant l'orientation et la progression du stage, une campagne expérimentale en partenariat avec le CEA-LETI pourra être lancée pour évaluer la pertinence des modèles de faute étudiés.
Ce sujet peut donner lieu à une continuation en thèse dans le cadre des projets nationaux PEPR SecurEval et Arsene, visant à développer des outils pour construire et évaluer des systèmes embarqués robustes aux attaques à l'état de l'art.

Profil recherché

Ce stage s'adresse à des candidat(e)s inscrit(e)s dans des formations en informatique, idéalement au niveau M2[1] (ou dernière année d'école d'ingénieur).
Le ou la candidat(e) devra justifier des compétences et connaissances suivantes :

- bonne capacités de programmation en C/C++
- connaissances solides en architecture des micro-processeurs

En outre, une appétence pour les questions liées à la sécurité informatique (sécurité matérielle, protection du code, ...) est vivement souhaitée. Cependant, le ou la candidat(e) pourra acquérir les compétences liées au cours du stage.

Candidatures

Pour candidater, envoyez un e-mail à pierre.corbineau@univ-grenoble-alpes.fr et bruno.ferres@univ-grenoble-alpes.fr avec votre CV, une courte lettre de motivation, et tout document pouvant étayer votre candidature.

Localisation

Le stage se déroulera au sein du laboratoire VERIMAG, sur le campus de Grenoble :

Laboratoire VERIMAG, Bâtiment IMAG,
150 place du Torrent,
38401 Saint-Martin-d'Hères

Bibliographie

[1] B. Colombier, A. Menu, J.-M. Dutertre, P.-A. Moëllic, J.-B. Rigaud, and J.-L. Danger, "Laser-induced single-bit faults in flash memory: Instructions corruption on a 32-bit microcontroller," in *2019 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pp. 1–10, IEEE, 2019.

[2] I. Alshaer, B. Colombier, C. Deleuze, V. Beroulle, and P. Maistri, "Variable-length instruction set: Feature or bug?," in *2022 25th Euromicro Conference on Digital System Design (DSD)*, pp. 464–471, IEEE, 2022.

[3] V. Werner, *Optimiser l'identification et l'exploitation de vulnérabilités à l'injection de faute sur microcontrôleurs*. PhD thesis, Université Grenoble Alpes, 2022.

[4] L. Dureuil, *Analyse de code et processus d'évaluation des composants sécurisés contre l'injection de faute*. PhD thesis, Université Grenoble Alpes, 2016.

[1] Les candidatures motivées au niveau M1 seront également étudiées