Sujet de thèse

Cross-Layer Fault Analysis for Microprocessor Architectures (CLAM)

LCIS, TIMA, Verimag

La sécurisation des éléments destinés au marché de l'IoT, ainsi que des infrastructures cyberphysiques critiques nécessite d'analyser leurs vulnérabilités, et de définir des contre-mesures matérielles et logicielles aux plus justes coûts. La complexité de plus en plus grande des processeurs et des applications qu'ils exécutent fait que les modèles de fautes logiciels (sauts d'instruction, remplacement d'instruction, etc) habituellement utilisés pour analyser la vulnérabilité de leur code ne suffisent plus à exprimer la diversité des comportements fautifs des architectures modernes [1]. En effet, les concepteurs d'architectures ont progressivement ajouté aux processeurs de nombreux blocs matériels complexes (pipeline, mémoire cache, prédiction de branchements, exécution spéculative, blocs spécialisés…) afin d'optimiser l'exécution des programmes. Parallèlement, les techniques d'injection de fautes ne cessent de progresser (attaques localisées ElectroMagnétique ou laser, attaques par injection de glitchs,...) autorisant aujourd'hui des injections multiples (à la fois multi-temporelle et multi-spatiales).

Face à ces attaques, les concepteurs doivent mettre en place des *contre-mesures* destinées à détecter ou masquer les effets des fautes injectées. Ces contre-mesures peuvent être mises en oeuvre au niveau matériel (duplication d'éléments, codes correcteurs d'erreur, mécanismes d'isolation, etc) ou au niveau logiciel (duplication d'instructions ou d'algorithmes, insertion de tests de sécurité, vérification de signatures, etc). La conception et la validation de ces contre-mesures nécessite de disposer d'une représentation réaliste des effets des attaques en faute (*modèle de fautes) au niveau matériel et/ou au niveau logiciel*. Les difficultés rencontrées sont alors : (1) la pertinence des modèles de fautes : représentent-ils correctement les effets qu'un attaquant peut engendrer sur un processeur cible? (2) l'adéquation des contre-mesures : les contre-mesures protègent-elles efficacement le système, ne sont-elles pas surdimensionnées et donc trop coûteuses? (3) la liaison entre les contre-mesures logicielles et matérielles : comment combiner efficacement et au coût le plus juste contre-mesures logicielles et matérielles, comment lier les différents niveaux pour la modélisation des fautes ?

Le projet CLAM et la thèse que nous proposons vise à répondre à ces questions en associant 3 laboratoires avec des spécialités et des expériences complémentaires : le LCIS (à Valence), spécialisé dans la simulation de fautes matérielles au niveau RTL et le développement d'outils d'injection de fautes (générateurs de glitchs d'horloge et de tension, attaque EM) ; le Laboratoire TIMA, avec son expertise dans l'évaluation, la modélisation et l'émulation des fautes ; le laboratoire Verimag spécialisé dans l'analyse des vulnérabilités logicielles à base d'outils d'analyse statique. Les travaux dans le domaine de l'injection de fautes de chacun de ces laboratoires sont reconnus nationalement et internationalement.

La figure 1 représente les différents niveaux d'analyse (processeur matériel cible, description RTL et code exécuté) et les outils que nous proposons de combiner pour créer de nouveaux modèles de fautes logiciels réalistes.

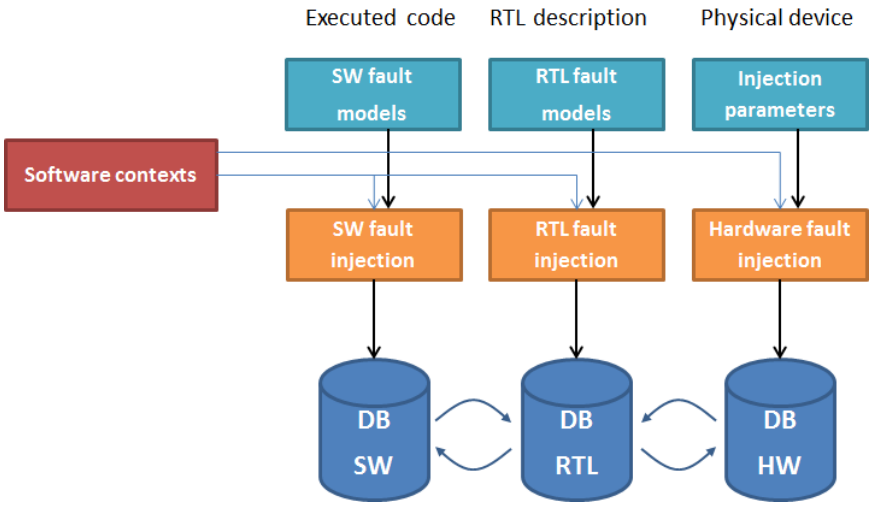

Figure 1. Plateforme d'injection de faute multi-niveau:
sur le composant physique, sur la description RTL et dans le code exécuté

Les attaques sur des cibles matérielles (processeurs ARM) permettront d'analyser le réalisme des fautes RTL et les fautes RTL d'analyser le réalisme des fautes logicielles. Par exemple, les fautes logicielles dont les effets ne sont jamais observées seront éliminées et à l 'inverse de nouvelles fautes logicielles seront créées. Grâce au nouveau modèle de fautes logiciel inféré pour chaque processeur et type d'attaques, le but est alors d'obtenir grâce aux outils d'analyse statique une analyse de vulnérabilités du code à la fois plus réaliste et plus rapide que ce qui se fait actuellement. Cette analyse permettra alors de vérifier et d'améliorer l'efficacité des contre-mesures proposées tant au niveau logiciel qu'au niveau matériel.

Profil du doctorant : Master en systèmes embarqués, Master en informatique, Master en microélectronique

Compétences : Compilation, Architecture des ordinateurs, Prototypage et simulation de systèmes numériques

Localisation : Grenoble INP LCIS, Valence

Contacts : vincent.beroulle@lcis.grenoble-inp.fr; paolo.maistri@univ-grenoble-alpes.fr

Pour candidater à cette offre merci d'envoyer aux personnes indiquées en contact : CV, lettre de motivation spécifique à ce sujet (en français ou anglais), bulletin de notes du master (M1 et M2), et lettres de recommandations

Références

[1] J. Laurent, V. Beroulle, C. Deleuze, F. Pebay-Peyroula, A. Papadimitriou, "Improving Software Fault Models and Countermeasures in a RISC-V Processor using a Cross-Layer Approach", Microprocessors and Microsystems, Elsevier, under publication 2019