Stage M2: Suivi dynamique de flux d'information dans les applications hybrides

Guillaume Hiet Pierre Wilke Frédéric Tronel

Contact: guillaume.hiet@centralesupelec.fr

1 Contexte

Parmi les différentes approches permettant de sécuriser les applications, le suivi dynamique des flux d'information (*Dynamic Information Flow Tracking*) permet de détecter un large spectre d'attaques contre l'intégrité et la confidentialité.

Le DIFT peut être implémenté à différents niveaux : au sein de l'OS [11], dans le code des applications [8] ou en modifiant le processeur [9]. Plus récemment, les approches *off-core* proposent d'implémenter le moniteur DIFT dans un co-processeur matériel dédié, afin de limiter les modifications apportées au cœur principal du CPU [4, 10].

L'isolation fournie par l'utilisation d'un co-processeur permet de protéger le moniteur DIFT, mais crée également un problème de fossé sémantique car le moniteur n'a pas accès directement au comportement des applications exécutées sur le cœur du processeur. Il est donc nécessaire de modifier le cœur principal pour qu'il fournisse les informations nécessaires au co-processeur [4] ou d'utiliser un mécanisme de trace [10] fourni par le processeur. Ce dernier type d'approche, que nous avons proposé dans le cadre du projet Labex CominLabs HardBlare, permet de s'affranchir des modifications sur le cœur principal et peut donc être utilisé pour protéger des applications exécutées sur des hardcore.

Toutefois, ces différents travaux de DIFT ne permettent pas de suivre les flux d'informations dans les applications hybrides, où une partie des traitements est déportée sur un FPGA. Quelques travaux se sont intéressés à vérifier les flux d'information d'un circuit au niveau HDL [1-3, 5]. Néanmoins, ces travaux s'intéressent exclusivement à la vérification statique de flux d'information au sein d'un circuit matériel. L'approche qui nous parait la plus prometteuse consiste à analyser et instrumenter le code intermédiaire d'un outil de type HLS (*High-Level Synthesis*) afin d'instrumenter le code HDL généré pour y insérer un moniteur DIFT spécifique au circuit [7]. Cependant, ces travaux portent principalement sur la génération des circuits matériels de propagation de tags correspondant aux blocs fonctionnels implémentés sur FPGA et les auteurs ne considèrent pas le cas où le DIFT des composants logiciels est lui aussi réalisé matériellement.

2 Objectifs

Dans le cadre du projet ANR TrustGW, qui commencera en janvier 2022, nous nous intéressons à un système composé d'objets connectés à une passerelle. Cette passerelle est à son tour connectée à un ou plusieurs serveurs dans le cloud. L'architecture de la passerelle au cœur du projet est hétérogène (logiciel / matériel). Elle est composée d'un processeur bande de base, un processeur applicatif, ainsi que des accélérateurs implémentés sur un FPGA.

Notre objectif est d'étendre le flot de conception matérielle afin de mettre en place les mécanismes de propagation des tags pour les accélérateurs matériels (implémentés sur FPGA, via HLS). Pour cela, nous souhaitons intégrer une phase d'analyse statique et d'instrumentation dans un outil de type HLS, afin de générer automatiquement des circuits de propagation de tags associés à chaque accélérateur. Ces circuits doivent également pouvoir échanger des tags avec le co-processeur DIFT en charge de la propagation des tags pour la partie logicielle des applications. Nous pensons par exemple utiliser des outils open-source comme Bambu [6]. Certains outils propriétaires comme Xilinx Vitis [1] permettent également d'insérer des modules d'analyse et d'optimisation dans leur chaîne de compilation.

Références

[1] A. Ardeshiricham, W. Hu, J. Marxen et R. Kastner. "Register transfer level information flow tracking for provably secure hardware design". In : *Design, Automation Test in Europe Conference Exhibition (DATE), 2017*. 2017, p. 1691-1696.

1. https://github.com/Xilinx/HLS/

[2] Armaiti Ardeshiricham, Yoshiki Takashima, Sicun Gao et Ryan Kastner. "VeriSketch : Synthesizing Secure Hardware Designs with Timing-Sensitive Information Flow Properties". In : *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. CCS '19. London, United Kingdom : Association for Computing Machinery, 2019, p. 1623-1638.

[3] Andrew Ferraiuolo, Rui Xu, Danfeng Zhang, Andrew C. Myers et G. Edward Suh. "Verification of a Practical Hardware Security Architecture Through Static Information Flow Analysis". In : *Proceedings of the Twenty-Second International Conference on Architectural Support for Programming Languages and Operating Systems*. ASPLOS '17. Xi'an, China : Association for Computing Machinery, 2017, p. 555-568.

[4] Hari Kannan, Michael Dalton et Christos Kozyrakis. "Decoupling dynamic information flow tracking with a dedicated coprocessor". In : *Dependable Systems & Networks, 2009*. IEEE. 2009, p. 105-114.

[5] Xun Li, Mohit Tiwari, Jason K. Oberg, Vineeth Kashyap, Frederic T. Chong, Timothy Sherwood et Ben Hardekopf. "Caisson : A Hardware Description Language for Secure Information Flow". In : *Proceedings of the 32nd ACM SIGPLAN Conference on Programming Language Design and Implementation*. PLDI '11. San Jose, California, USA : Association for Computing Machinery, 2011, p. 109-120.

[6] C. Pilato et F. Ferrandi. "Bambu : A modular framework for the high level synthesis of memory-intensive applications". In : *2013 23rd International Conference on Field programmable Logic and Applications*. 2013, p. 1-4.

[7] C. Pilato, K. Wu, S. Garg, R. Karri et F. Regazzoni. "TaintHLS : High-Level Synthesis for Dynamic Information Flow Tracking". In : *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 38.5 (2019), p. 798-808.

[8] Feng Qin, Cheng Wang, Zhenmin Li, Ho-seop Kim, Yuanyuan Zhou et Youfeng Wu. "LIFT : A Low-Overhead Practical Information Flow Tracking System for Detecting Security Attacks". In : *Proceedings of the 39th Annual IEEE/ACM International Symposium on Microarchitecture*. MICRO 39. USA : IEEE Computer Society, 2006, p. 135-148.

[9] G Edward Suh, Jae W Lee, David Zhang et Srinivas Devadas. "Secure program execution via dynamic information flow tracking". In : *Acm Sigplan Notices*. T. 39. 11. ACM. 2004, p. 85-96.

[10] M. A. Wahab, P. Cotret, M. N. Allah, G. Hiet, V. Lapôtre et G. Gogniat. "ARMHEx : A hardware extension for DIFT on ARM-based SoCs". In : *27th International Conference on Field Programmable Logic and Applications (FPL)*. 2017.

[11] Nickolai Zeldovich, Silas Boyd-Wickizer, Eddie Kohler et David Mazières. "Making Information Flow Explicit in HiStar". In : *Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation - Volume 7*. OSDI '06. Seattle, WA, 2006, p. 19-19.