Plateforme d'analyse de signaux micro-architecturaux pour la mise en place de compteurs matériels de sécurité

1st Lucas Georget
EDF R&D / LAAS-CNRS
Palaiseau / Toulouse, France
lucas.georget@{edf/laas}.fr

2nd Vincent Migliore
LAAS-CNRS
Toulouse, France
vincent.migliore@laas.fr

3rd Vincent Nicomette
LAAS-CNRS
Toulouse, France
vincent.nicomette@laas.fr

4th Frédéric Silvi
EDF R&D
Palaiseau, France
frederic.silvi@edf.fr

5th Arthur Villard
EDF R&D
Palaiseau, France
arthur.villard@edf.fr

***Abstract*—La détection de comportements malveillants logiciels ou matériels pendant le fonctionnement d'un système informatique demande la mise en place d'observables provenant d'une ou plusieurs couches d'abstraction de ce dernier. Cette abstraction cependant tend à limiter la capacité à détecter des déviations de comportement, surtout pour des classes d'attaques qui exploitent des vulnérabilités très proches du matériel cible. A contrario, un niveau d'abstraction trop faible tend à faire croître significativement la complexité du modèle du système et donc pose un certain nombre de difficultés pour l'extraction et la sélection des observables pertinents pour une classe d'attaque donnée. Les compteurs de performance des processeurs ont notamment été utilisés comme moyen indirecte d'observer le comportement de la micro-architecture et détecter des logiciels tentant d'exploiter des vulnérabilités matérielles. Afin d'améliorer les différentes méthodes de détection, nous proposons la construction de métriques matérielles pensées dès la conception pour la sécurité en étudiant la corrélation entre les signaux provenant de la micro-architecture et les différentes classes d'attaque de la littérature ciblant à la fois des systèmes IT classiques et OT industriels. Par extension, ces travaux ont pour ambition de détecter des attaques provenant de chevaux de Troie matériels, ces derniers ayant pour effet de changer le comportement d'une micro-architecture donnée.**

***Index Terms*—hardware security counters, runtime attack detection, microarchitectural signals analysis, FPGA SoC, hardware trojan insertion**

I. Introduction

Dans l'économie mondialisée d'aujourd'hui, le cycle de vie des composants dépend de nombreux facteurs. Ceci est particulièrement important pour la conception des circuits intégrés, qui sont essentiels à l'électronique. Que ce soit au niveau de la conception ou de la fabrication, le recours à la sous-traitance peut poser des problèmes de souveraineté. Avec les Chips Acts [1], de grands groupes promeuvent actuellement une plus grande indépendance des semi-conducteurs, et des universitaires militent pour une connaissance plus ouverte des différents outils nécessaires pour y parvenir [2]. Des travaux très intéressants sont également menés sur la protection contre la contrefaçon [3] et la préservation de la propriété intellectuelle de leurs technologies tant qu'elles sont produites à l'extérieur de l'entreprise. Par ailleurs, des organismes tels que les Centres d'Évaluation de la Sécurité des Technologies de l'Information (CESTI) [4] sont chargés de vérifier un certain nombre de propriétés de sécurité pour les composants que les entreprises souhaitent intégrer dans leur infrastructure. La fuite de données confidentielles ou le déni de problème découlant d'événements dans la chaîne d'approvisionnement pose un réel problème de confiance entre les différents acteurs de l'environnement numérique. Pour donner quelques exemples, un sous-traitant en charge du développement d'un programme binaire peut livrer un binaire corrompu à cause d'un compilateur malveillant [5] installé dans le système d'information du sous-traitant, ou à cause d'un conteneur corrompu utilisé pour livrer une application [6]. Ces corruptions peuvent même viser le matériel, avec des implémentations radio modifiées [7], ou des chevaux de Troie. Les corruptions matérielles doivent être prises très au sérieux car elles sont plus difficiles à détecter et à corriger que les corruptions logicielles.

Nos travaux de recherche ont pour objectif de détecter de telles corruptions matérielles. Nous considérons un scénario de menace dans lequel nous excluons une présence physique d'un attaquant au sein de l'infrastructure critique à protéger, le contrôle et la gestion des accès étant efficace. Les systèmes que nous considérons sont des automates de contrôle-commande, qui sont, pour les parties sensibles, isolés des réseaux extérieurs, c'est pourquoi nous ne nous concentrons pas sur les attaques à distance. Nous considérons que, lors des différentes étapes de cycle de vie d'un composant, des vulnérabilités peuvent être intentionnellement introduites. A titre d'exemple, le schéma Figure 1 reprend les niveaux de confiance accordés à chaque phase dans la chaîne d'approvisionnement d'un circuit intégré vis-à-vis de la menace d'insertion de chevaux de Troie matériel.

Par ailleurs, les industriels ont de plus en plus le besoin de s'affranchir de solutions techniques matérielles propriétaires en boîte noire, ou de problèmes d'obsolescence d'ancien matériel que l'on ne trouve plus sur étagère. Pour cela une piste intéressante est l'utilisation de matériel reconfigurable (qui pourra être patché/mis à jour par la suite). Néanmoins les délais de classification en sûreté de fonctionnement pour les systèmes industriels peuvent être très longs. De plus, l'utilisation des langages de description matérielle (HDL)

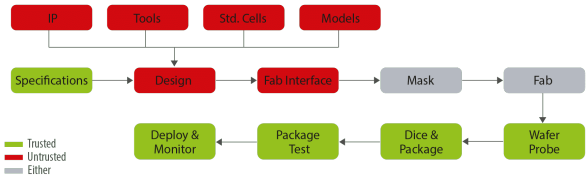

Fig. 1. Hardware Trojans' threat in IC supply chain [8]

introduit également de nouveaux scénarios de menace de chevaux de Troie matériels [9]. Et comme les flux binaires sont propriétaires, ils ne peuvent donc pas être retracés et analysés facilement pour remonter aux informations qu'ils contiennent, même si des travaux sont en cours à ce sujet [10]. Des garanties peuvent donc être demandées sur le contenu du code, ou la vérification d'une empreinte de firmware, mais il est difficile d'avoir confiance dans l'ensemble de la chaîne de validation.

L'ensemble de ces menaces, dans un contexte de plus en plus complexe, ou logiciel et matériel sont en forte interaction, où les architectures matérielles reconfigurables s'imposent de plus en plus, il est donc fondamental de pouvoir être en capacité de détecter des attaques, notamment matérielles, avec des mécanismes adaptés, au bon niveau d'abstraction. Les signaux de la micro-architecture constituent pour cela un bon candidat mais il est très difficile d'identifier les signaux pertinents pour détecter une attaque spécifique. Une plateforme de capture et d'analyse de signaux micro-architecturaux, permettant de mener des expérimentations variées est un prérequis fondamental. Cet article décrit une telle plateforme d'expérimentation, dont l'objectif est d'aider à la construction de métriques matérielles spécifiques pour la sécurité en étudiant la corrélation entre signaux micro-architecturaux et différentes classes d'attaques.

La section II donne un rapide panorama de l'état de l'art. La section III décrit ensuite la plateforme que nous avons conçue afin de permettre l'analyse de signaux micro-architecturaux pour la définition de compteurs matériels de sécurité. La Section IV propose enfin quelques perspectives à ces travaux.

II. Etat de l'art

A. Chevaux de Troie matériels

Parmi les attaques visant les composants matériels, les chevaux de Troie matériels sont particulièrement redoutables. Ils ne requièrent aucune présence physique, aucune connexion ou simplement aucune action de la part de l'attaquant. Nos recherches se concentrent sur ce type spécifique d'attaques. Il s'agit d'altérations du matériel à tous les stades possibles de la vie d'un composant, dans le but de modifier son fonctionnement normal. Sur la base d'une taxonomie complète [11] réalisée en 2010, des benchmarks de chevaux de Troie ont été réalisés en 2013 [12], et mis à jour en 2017 [13]. Les conséquences de ces chevaux de Troie peuvent aller de la fuite de données sensibles au déni de service, voire à la destruction de la puce [14]. Un autre point important est qu'ils peuvent être activés dès le départ, "toujours actifs", ou déclenchés par un événement rare. Ces variations modifient les mécanismes de détection à appliquer. Il est même possible d'avoir des scénarios dans lesquels les chevaux de Troie peuvent interagir à distance [15], en particulier par radio. Les chevaux de Troie peuvent être combinatoires, séquentiels ou analogiques [16], ce qui modifie l'approche à adopter pour les observer. Diverses méthodes de détection existent actuellement dans la littérature, chacune avec ses avantages et ses inconvénients.

B. Hardware Performance Counters et Instruction Flow Tracing

Les Hardware Performance Counters (HPC) ont été utilisés de nombreuses fois pour des besoins de sécurité. Des premiers travaux [17] ont, dans le contexte d'une flotte d'appareil IoT avec les mêmes fonctions, cherché à identifier les déviations de comportement d'un ou plusieurs appareils par rapport au fonctionnement des autres. Ensuite, un système hybride [18] avec une partie locale et analyse partielle, et une remontée de données pour analyse via Machine Learning/IA si seuil malveillant détecté. Par ailleurs, différentes méthodes reposant sur des algorithmes d'apprentissages ont été utilisés sur les HPCs pour détecter des attaques temporelles sur les caches de processeur ont été recensés dans la thèse de Maria Mushtaq [19]. A nouveau ces travaux ont permis la réalisation d'un système de surveillance et traçage pour les systèmes légers [20] face à des attaques radios. De plus, pour le cas d'application des chevaux de Troie matériel, deux travaux de Elnaggar et al [21],[22] utilisent les compteurs de performance en corrélation avec l'activité du processeur pour les détecter. Cependant ce sont pour des attaques plutôt brusques, qui impactent fortement le déroulement des instructions. D'autres travaux [23] qui s'approchent de la micro-architecture permettent de modéliser l'activité du prédicateur de branchement et des accès en cache afin de repérer les opérations illégitimes, mais cela suppose une quantité d'information et de confiance non négligeable pour les modélisations en amont.

C. Hardware Signal Probing

Pour atteindre un niveau de granularité très fin afin de détecter des attaques subtiles, une solution est de revenir aux signaux directement observés par les compteurs de performance afin d'en reconstruire des compteurs spécifiques adaptés aux besoins de la détection. Pour le moment, seules des solutions de debug tels que Xilinx ChipScope et Intel SignalTap sont présentes dans les outils du marché. Mao & al[24] ont travaillé sur une instrumentation d'un RocketChip en Scala afin d'apporter une telle solution face aux attaques logicielles repérables uniquement au niveau micro-architectural, avec des contraintes fortes sur le nombre de données remontées. Concernant les chevaux de Troie, Hou et al [25] proposent pour un ASIC de regarder une liste de signaux prédéfinis afin de détecter un cheval de Troie analogue de référence très subtil, le cheval de Troie A2 [26].

III. Plateforme d'analyse de signaux micro-architecturaux pour la détection d'intrusion

A. Vue globale

Le travail que nous menons dans cette thèse se situe dans la poursuite des travaux de recherche décrits dans la dernière partie de l'état de l'art, concernant l'analyse des signaux de la micro-architecture. Pour cela, la mise en place d'une plateforme d'analyse des signaux micro-architecturaux capable de traiter avec efficacité une grande quantité de données à remonter est nécessaire. Plus précisément, les objectifs de cette plateforme sont :

- de proposer un cadre amélioré de surveillance et d'analyse des signaux micro-architecturaux au moment de l'exécution, pour mieux comprendre les interactions entre le logiciel et la micro-architecture au moment de l'exécution
- de faciliter la mise en place d'une méthodologie, à partir de l'analyse de signaux micro-architecturaux choisis à dessein, de détecter diverses familles d'attaques via la caractérisation de leurs empreintes micro-architecturales
- de permettre des expérimentations dans un contexte open source sur du matériel réel avec le noyau Linux
- de faciliter la présentation et l'analyse approfondies des cas d'utilisation dans lesquels nous avons construit avec succès une logique de détection appropriée pour différentes corrélations d'attaques via des compteurs de sécurité matériels.

Concrètement, pour réaliser de telles expérimentations, nous avons besoin de :

- Un système cible, reconfigurable, avec un analyseur logique intégré qui permet l'extraction de signaux micro-architecturaux
- Un système hôte qui collecte ces données, avec donc une bonne capacité de stockage et une bonne bande passante avec la cible
- Un système performant (peut-être le même que l'hôte) pour traiter ensuite les données et les analyser.

Pour se faire nous utilisons pour la cible une carte Alveo U50-DD [27], avec 8 Go de mémoire HBM (rapide), connectée en PCIe x4 (x16 possible) avec le système hôte, un ordinateur fixe avec plusieurs To de mémoire de disponibles.

Afin de paramétrer la plateforme, nous utilisons LiteX [28], qui est un moyen de construire facilement des systèmes sur puce, sur des cartes FPGAs. Après avoir porté la carte sur le projet, nous avons pu faire exécuter un petit système Linux.

Pour la partie analyseur logique embarqué, LiteScope [29] a été intégré afin d'avoir une première observation des signaux micro-architecturaux provenant par exemple des bus d'instructions et de données dans le CPU.

B. Observation des signaux

Plusieurs options sont disponibles pour l'observation des signaux micro-architecturaux dans le framework. On peut observer l'ensemble des traces du système sur puce, uniquement

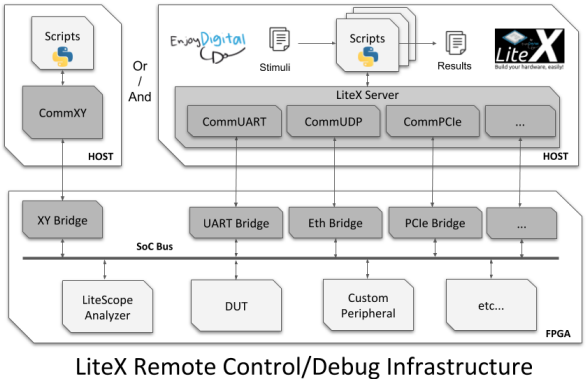

Fig. 2. Use Host Bridge to control debug a SoC

observer l'activité interne du microprocesseur, ou uniquement observer l'activité modules pour les liaisons périphériques par exemple. A la manière des méthodes de déboggage en développement matériel, il est possible de laisser des sondes nous remonter la valeur des signaux transitant sur les différentes parties de notre puce FPGA via un analyseur logique intégré.

Actuellement, la simulation avec Verilator permet de monitorer l'ensemble du trafic de tous les signaux provenant de notre système reconfigurable, mais cela représente plusieurs dizaines de Go de données pour quelques minutes d'utilisation d'un BIOS sans grande activité. Cela montre qu'un choix préalable des différents observables à sélectionner va devoir être fait, et expérimenter, afin de pouvoir traiter ces données efficacement dans un temps et une quantité d'énergie raisonnables.

Lors de l'utilisation initiale de LiteScope, il est par exemple proposé de regarder l'activité du CPU VexRiscv au niveau des bus de données et d'instruction. En adaptant cela en fonction de la quantité de mémoire disponible, car plus de signaux sont observés en parallèle, plus vite cette dernière se remplit et sature, l'analyse d'autres CPUs aux architectures différentes peut être envisagée. Il est également possible d'observer d'autres signaux que les bus, hors ou dans le CPU.

L'outil est encore en développement, la sous-section III-D détaille les travaux en cours à ce sujet pour notre problématique sur la plateforme.

Les communications entre le système hôte et la carte FPGA se font via UART à travers le PCIe dans notre cas, ce qui sera grandement utile lorsqu'une quantité importante de données seront remontées en continu. La mise en place de la plateforme a été pensée pour permettre une transmission efficace des données utiles à nos expérimentations.

C. Cas d'usages

Les premiers cas d'études que nous souhaitons expérimenter avec ce type de plateforme, concernent principalement deux catégories d'attaques :

- Des attaques d'origine logicielle telles que les Cache Side-Channel et Return-Oriented Programming Attacks,

mais aussi possiblement Spectre, Meltdown et Rowhammer
- Des attaques matérielles telles qu'insertion automatique ou manuelle de chevaux de Troie au niveau du processeur et des périphériques.

Les métriques collectées durant ces différentes expérimentations seront stockées et analysées (avec du traitement via Machine Learning notamment) afin d'identifier des signaux pertinents pour créer des compteurs matériels de sécurité. Les compteurs pourront être basés sur des seuils de certains signaux spécifiques et probablement sur la corrélation de valeurs de plusieurs signaux selon les différentes classes d'attaques.

Cela pourrait aussi permettre de caractériser le matériel et d'en déduire son comportement usuel pour faire de la détection d'anomalies plutôt que de signatures d'attaques pour des architectures peu complexes, ce qui se prête bien à des scénarios industriels.

D. Travaux en cours

Afin de finaliser la plateforme pour nos cas d'usages, nous travaillons actuellement à :

- Utiliser des images OS customs afin d'exécuter les benchmarks de programme bénins et malveillants. Pour le moment un petit système d'exploitation s'exécute sur la plateforme mais avec peu de ressources et de fonctionnalités. Il serait intéressant de pouvoir exécuter des systèmes d'exploitation plus riches en fonctionnalités.
- Modifier LiteScope pour réaliser du monitoring continu, car actuellement le monitoring reste limité à un espace de stockage prédéfini qui ne permet de couvrir qu'une période limitée de temps d'utilisation du système, selon la quantité de mémoire pouvant être empruntée par le module, et l'activité des signaux
- Réaliser la segmentation et la corrélation des relevés de signaux par processus, de manière à ne prendre en compte que les changements dûs au programme lancé (dans le cadre d'une architecture mono-coeur, sans interruption).

IV. Ouvertures

Ces travaux de détection d'attaques logicielles impactant la micro-architecture, ou de chevaux de Troie matériels, pourraient être amenés à s'étendre à des fins de rétro-ingénierie ou d'investigation (forensic) au niveau du matériel. Les compteurs de sécurité peuvent être stockés en interne ou exportés, afin de garder des traces (logs) des activités passés, et/ou permettre la caractérisation de l'interaction logiciel/matériel de manière à identifier l'architecture utilisée ainsi que les programmes s'exécutant sur cette architecture. Cela apporterait aussi par la suite des possibilités de diagnostic de la défaillance, voire de remédiation si techniquement la reconfiguration partielle est activée.

Pour la partie industrielle, les techniques déjà présentes en sûreté de fonctionnement (redondance, vote majoritaire, ...) [30] trouvent leur application pour apporter des contre-mesures efficaces face aux chevaux de Troie matériels, et autres défaillances volontaires ou non. Des attaques pourraient être pensées spécialement pour ces mécanismes, afin de réfléchir aux futurs moyens de s'en prémunir. De même des capacités de remédiation spécifique pour ces équipements pourraient être imaginées dans la mesure où les systèmes critiques ne peuvent être interrompus et doivent fonctionner en comportement dégradé éventuellement.

En outre, des parallèles seraient intéressants aussi avec les couches supérieures, pour de la transmission radio ou réseau. Des attaques venant de ces dernières viendrait perturber le comportement légitime du système sans pour autant utiliser d'opérations illégitimes. L'étape de caractérisation du matériel serait donc encore plus importante pour ne pas laisser passer ce genre de malveillances.

Enfin, une réflexion sur la confiance accordée tout au long de la chaîne d'approvisionnement, ainsi que lors du déploiement et mise en service du matériel, doit être menée sur la suite des travaux. Des systèmes autonomes sur la base des travaux de Ken Thompson [5] ont vu le jour récemment [31] pour les parties matérielles, et se pose la question de malwares discrets à ce niveau-là pouvant être présents sur ces derniers.

References

[1] *European chips act*, https://commission.europa.eu/strategy-and-policy/priorities-2019-2024/europe-fit-digital-age/european-chips-act_en [Accessed: 05/07/2024], 2023.

[2] L. Benini, G. D. Micheli, M.-M. Louërat, H. Pretl, and S. Wallentowitz, *Importance of open-source eda tools for academia*, https://open-source-eda-letter.eu [Accessed: 05/07/2024], 2024.

[3] S. Dupuis and M.-L. Flottes, "Logic Locking: A Survey of Proposed Methods and Evaluation Metrics," *Journal of Electronic Testing: : Theory and Applications*, vol. 35, no. 3, pp. 273–291, May 2019. DOI: 10.1007/s10836-019-05800-4. [Online]. Available: https://hal-lirmm.ccsd.cnrs.fr/lirmm-02128826.

[4] ANSSI, *Security certification of products*, https://cyber.gouv.fr/sites/default/files/2022-08/security-certification-of-products_security_visa_anssi\%5B1\%5D.pdf [Accessed: 05/13/2024].

[5] K. Thompson, "Reflections on trusting trust," *Commun. ACM*, vol. 27, no. 8, pp. 761–763, Aug. 1984, ISSN: 0001-0782. DOI: 10.1145/358198.358210. [Online]. Available: https://doi.org/10.1145/358198.358210.

[6] F. Moriconi, A. I. Neergaard, L. Georget, S. Aubertin, and A. Francillon, "Reflections on trusting docker: Invisible malware in continuous integration systems," in *WOOT 2023, 17th IEEE Workshop on Offensive Technologies, co-located with IEEE S&P 2023, 25 May 2023, San Francisco, United States*, IEEE, Ed., © 2023 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to

servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE., San Francisco, 2023.

[7] R. Cayre, F. Galtier, G. Auriol, V. Nicomette, M. Kaâniche, and G. Marconato, "WazaBee: attacking Zigbee networks by diverting Bluetooth Low Energy chips," in *IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2021)*, Taipei (virtual), Taiwan, Jun. 2021. DOI: 10.1109/DSN48987.2021.00049. [Online]. Available: https://laas.hal.science/hal-03193299.

[8] Secure-IC, *Detecting and protecting from hardware trojans*, 2021. [Online]. Available: https://www.secure-ic.com/applications/challenges/hardware-trojans/.

[9] J. Cruz, Y. Huang, P. Mishra, and S. Bhunia, "An automated configurable trojan insertion framework for dynamic trust benchmarks," in *2018 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2018, pp. 1598–1603. DOI: 10.23919/DATE.2018.8342270.

[10] S. Kashani, M. Emami, and J. R. Larus, "Bitfiltrator: A general approach for reverse-engineering xilinx bitstream formats," in *2022 32nd International Conference on Field-Programmable Logic and Applications (FPL)*, 2022, pp. 01–08. DOI: 10.1109/FPL57034.2022.00039.

[11] M. Tehranipoor and F. Koushanfar, "A survey of hardware trojan taxonomy and detection," *Design & Test of Computers, IEEE*, vol. 27, pp. 10–25, Mar. 2010. DOI: 10.1109/MDT.2010.7.

[12] H. Salmani, M. Tehranipoor, and R. Karri, "On design vulnerability analysis and trust benchmarks development," in *2013 IEEE 31st International Conference on Computer Design (ICCD)*, 2013, pp. 471–474. DOI: 10.1109/ICCD.2013.6657085.

[13] B. Shakya, T. He, H. Salmani, D. Forte, S. Bhunia, and M. Tehranipoor, "Benchmarking of hardware trojans and maliciously affected circuits," in *Journal of Hardware and Systems Security*, 2017. DOI: 10.1007/s41635-017-0001-6.

[14] M. Xue, C. Gu, W. Liu, S. Yu, and M. O'Neill, "Ten years of hardware trojans: A survey from the attacker's perspective," *IET Computers & Digital Techniques*, vol. 14, pp. 231–246, Sep. 2020. DOI: 10.1049/iet-cdt.2020.0041.

[15] A. Moschos, K. Valakuzhy, and A. D. Keromytis, "On the feasibility of remotely triggered automotive hardware trojans," in *2022 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, 2022, pp. 1–6. DOI: 10.1109/ICECCME55909.2022.9987857.

[16] A. Jain, Z. Zhou, and U. Guin, "Survey of recent developments for hardware trojan detection," in *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2021, pp. 1–5. DOI: 10.1109/ISCAS51556.2021.9401143.

[17] M. Bourdon, P.-F. Gimenez, E. Alata, *et al.*, "Hardware-performance-counters-based anomaly detection in massively deployed smart industrial devices," in *2020 IEEE 19th International Symposium on Network Computing and Applications (NCA)*, 2020, pp. 1–8. DOI: 10.1109/NCA51143.2020.9306726.

[18] N. F. Polychronou, P.-H. Thevenon, M. Puys, and V. Beroulle, "Madman: Detection of software attacks targeting hardware vulnerabilities," in *2021 24th Euromicro Conference on Digital System Design (DSD)*, 2021, pp. 355–362. DOI: 10.1109/DSD53832.2021.00060.

[19] M. Mushtaq, "Software-based Detection and Mitigation of Microarchitectural Attacks on Intel's x86 Architecture," Theses, Université de Bretagne Sud, Sep. 2019. [Online]. Available: https://theses.hal.science/tel-02988980.

[20] M. El-Bouazzati, "A Lightweight Host-based Intrusion Detection System using a Hardware-Assisted Monitor to detect Wireless Attacks Targeting Constrained IoT Devices," Theses, Université de Bretagne Sud, Dec. 2023. [Online]. Available: https://cnrs.hal.science/tel-04612764.

[21] R. Elnaggar, K. Chakrabarty, and M. B. Tahoori, "Run-time hardware trojan detection using performance counters," in *2017 IEEE International Test Conference (ITC)*, 2017, pp. 1–10. DOI: 10.1109/TEST.2017.8242063.

[22] R. Elnaggar, K. Chakrabarty, and M. B. Tahoori, "Hardware trojan detection using changepoint-based anomaly detection techniques," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 12, pp. 2706–2719, 2019. DOI: 10.1109/TVLSI.2019.2925807.

[23] B. Amornpaisannon, A. Diavastos, L.-S. Peh, and T. E. Carlson, "Secure run-time hardware trojan detection using lightweight analytical models," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 43, no. 2, pp. 431–441, 2024. DOI: 10.1109/TCAD.2023.3316113.

[24] Y. Mao, V. Migliore, and V. Nicomette, "Matana: A reconfigurable framework for runtime attack detection based on the analysis of microarchitectural signals," *Applied Sciences*, vol. 12, no. 3, 2022, ISSN: 2076-3417. DOI: 10.3390/app12031452. [Online]. Available: https://www.mdpi.com/2076-3417/12/3/1452.

[25] Y. Hou, H. He, K. Shamsi, Y. Jin, D. Wu, and H. Wu, "On-chip analog trojan detection framework for microprocessor trustworthiness," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 38, no. 10, pp. 1820–1830, 2019. DOI: 10.1109/TCAD.2018.2864246.

[26] K. Yang, M. Hicks, Q. Dong, T. Austin, and D. Sylvester, "A2: Analog malicious hardware," in *2016 IEEE Symposium on Security and Privacy (SP)*, 2016, pp. 18–37. DOI: 10.1109/SP.2016.10.

[27] AMD, *Amd alveo™ u50 data center accelerator card*, 2023. [Online]. Available: https://www.xilinx.com/publications/product-briefs/alveo-u50-product-brief-v2.pdf.

[28] F. Kermarrec, S. Bourdeauducq, J.-C. L. Lann, and H. Badier, *Litex: An open-source soc builder and library based on migen python dsl*, 2020. arXiv: 2005.02506 [cs.AR]. [Online]. Available: https://arxiv.org/abs/2005.02506.

[29] EnjoyDigital, *Litescope - a small footprint and configurable embedded fpga logic analyzer*, 2015. [Online]. Available: https://github.com/enjoy-digital/litescope.

[30] H. A. Amin, Y. Alkabani, and G. M. Selim, "System-level protection and hardware trojan detection using weighted voting," *Journal of Advanced Research*, vol. 5, no. 4, pp. 499–505, 2014, Cyber Security, ISSN: 2090-1232. DOI: https://doi.org/10.1016/j.jare.2013.11.008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2090123213001446.

[31] G. Somlo, *Self-hosting (almost) all the way down*, 2023. [Online]. Available: https://archive.fosdem.org/2023/schedule/event/rv_selfhosting_all_the_way_down.