Poster: A microarchitectural signals analysis platform to craft Hardware Security Counters

Lucas Georget[1,2], Vincent Migliore[1], Vincent Nicomette[1], Arthur Villard[2], and Frédéric Silvi[2]

[1] LAAS-CNRS, Toulouse, France surname.name@laas.fr
https://www.laas.fr/en/teams/trust/
[2] EDF R&D, Paris-Saclay, France surname.name@edf.fr

Abstract. Detecting malicious software or hardware behavior during the operation of a computer system requires observables from one or more abstraction layers of the system. However, this abstraction tends to limit the ability to detect behavioral deviations, especially for attack classes that exploit vulnerabilities very close to the target hardware. Conversely, too low a level of abstraction tends to significantly increase the complexity of the system model, and therefore poses a number of difficulties for the extraction and selection of relevant observables for a given class of attack. In particular, processor performance counters have been used as an indirect means of observing microarchitecture behavior and detecting software attempting to exploit hardware vulnerabilities. In order to improve the various detection methods, we propose the construction of hardware metrics designed from the outset for security, by studying the correlation between signals from the microarchitecture and the various classes of attack in the literature, targeting both usual and industrial systems. By extension, this work aims to detect attacks originating from hardware Trojans, the latter having the effect of changing the behavior of a given microarchitecture.

Keywords: Hardware Security Counters · Runtime attack detection · Microarchitectural signal analysis.

1 Introduction

The detection of low-level attacks, especially hardware malware or software malware targeting microarchitectural vulnerabilities, is quite complex. Depending on the System on Chip (SoC) architecture it could turn out very differently. Hardware Performance counters (HPC) for example, can be used to trace hardware behavior and divert it for security purposes. However, the further down we go, the more complex it becomes to use observables. No hardware metrics were originally designed for security purpose.

In an increasingly complex context, where software and hardware are in close interaction, and where reconfigurable hardware architectures are becoming more and more prevalent, it is essential to be able to detect attacks with appropriate mechanisms, at the right level of abstraction. Microarchitecture signals are a good candidate for this, but it is very difficult to identify the relevant signals for detecting a specific attack. This is why it is essential to study the impact of attacks on microarchitectural signals in order to build specific counters that could be used to reference the internal state of our machines and detect attacks at microarchitectural level, as well as hardware Trojans, on both traditional and industrial equipment. A platform for capturing and analyzing microarchitectural signals, enabling a variety of experiments to be carried out, is a fundamental prerequisite. This article describes such an experimental platform, which facilitates the building of specific hardware metrics for security by studying the correlation between microarchitectural signals and different classes of attack.

The 2 section gives a quick overview of the state of the art. Section 3 then describes the platform we have designed to enable the analysis of microarchitectural signals for the definition of security hardware counters. Section 4 finally proposes some perspectives to this work.

2 Related Work

2.1 Hardware Performance Counters

Hardware Performance Counters (HPC) have been used many times for security purposes. Early work [1], in the context of a fleet of IoT devices executing the same application, sought to identify deviations in the behavior of one or more devices compared to the others. An hybrid intrusion detection system [7] was created, by means of a local analysis on the devices themselves, along with a global analysis through machine learning algorithms on a remote server, in order to identify outliers in the fleet of devices. Some other research works based on learning algorithms have used HPCs to detect timing attacks on processor caches, as reported in Maria Mushtaq's thesis [6]. Against radio attacks, research works proposed the development of a monitoring and tracing system for lightweight systems [2].

2.2 Hardware Signal Probing

In the industrial context, it is not possible to run additional software on the system for command and control systems, so we need to find a way of passively collecting data directly in the hardware. Similarly, for fast attacks requiring immediate action, the response time is too long due to the number of clock cycles required for context switching. To reach a finer level of granularity and detect even more subtle attacks, it is necessary to analyze various hardware signals and try to identify which signals are relevant to detect some specific class of attacks. This will simplify the number of measures required, as they will be directly focused on security and require less correlation. Some specific platforms, mainly based on FPGA, are necessary to carry out such experiments. But, to the best of our knowledge, few solutions are currently available for that purpose. At present, only debugging solutions such as Xilinx ChipScope and Intel SignalTap are available on the market. Mao et al [5] have been working on the instrumentation of a RocketChip in Scala to provide such a solution for software attacks that can be detected at the microarchitectural level. Directly probing the signals associated with attack classes/families for security purposes enables them, with only very simple heuristics, to detect these attacks quickly and thus build counters with the different thresholds for each metric. But this work comes with severe constraints on the amount of data that can be retrieved.

The purpose of our research work is thus to design and implement a hardware platform, generic enough so that we can observe and exports various microarchitectural signals from the board (processor and peripherals) to perform a concrete and complete analysis and correlate this data.

3 Microarchitectural signals monitoring platform

3.1 Global view

The microarchitectural signal analysis platform should be capable of efficiently process large quantities of data. Our objective is to propose a flexible solution that captures a vector of internal signals from a System on Chip's microarchitecture with no impact on the running software. Concretely, to carry out such experiments, the solution requires:

- A reconfigurable target system, with an integrated logic analyzer for extracting microarchitectural signals.
- A host system that collects this data, with good storage capacity and good bandwidth with the target
- A high-performance system (perhaps the same as the host) to further process and analyze the data.

The solution has been implemented on the high-end FPGA-based plateform Alveo U50-DD connected to a host computer through PCIe, for high data transfer speed capacity.

For a fast and portable SoC deployment into the programmable logic, we used the emerging framework LiteX [4], which is an easy way of building systems-on-a-chip, on FPGA boards. After porting the board to the project, we were able to run a small Linux system on it.

For the on-board logic analyzer, LiteScope [3] has been integrated into the SoC as described in Figure 1 to provide a view of the microarchitectural signals. For example, it allows to observe the instruction and data buses on a VexRiscv CPU. It is a small footprint and configurable tool able to capture signals in real time, with limited resources and without any perturbation of the system. It can be customized for our needs.

3.2 Use-cases / Construction of Hardware Security Counters

The first case studies we want to experiment with this type of platform mainly concern two categories of attack:

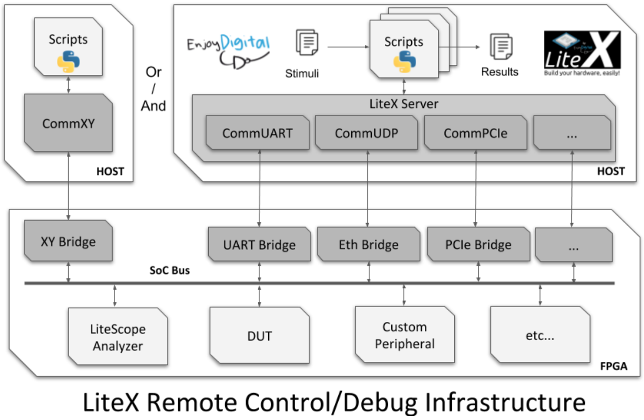

Fig. 1. Use Host Bridge to control debug a SoC [3][4]

- Software attacks, such as Cache Side-Channel and Return-Oriented Programming attacks, but also possibly Spectre, Meltdown and Rowhammer.
- Hardware attacks such as automatic or manual insertion of Trojans at processor and peripheral level.

The metrics collected during these various experiments will be stored and analyzed (with processing via Machine Learning in particular) in order to exhibit common detection criteria that will be used to design relevant hardware security counters. The counters could be based on thresholds of specific signals, and probably on the correlation of values of several signals according to different classes of attack. According to the scenarios, on light architectures for the industry or complex ones for computer systems, it will be possible to have different characterization proper to each hardware.

4 Ongoing and future work

We're currently testing our platform with different RISC-V softcore implementations, that can run operating systems such as a classical Linux, or a real-time Zephyr, depending on the needs. By means of LiteScope, the monitoring of the peripheral buses (connected to main SoC's bus) has been realized on a VexRiscv CPU. It is possible to observe the different memory accesses on a Rocket CPU, such as the AXI Memory (L1-cached), AXI MMIO (not cached), AXI L2FB (Slave, for e.g., DMA).

This platform can be used for instance to improve previous work ([5]) focusing on the processor core for software attacks, by extracting relevant information relative to the instructions and memory accesses. At the moment, we only extract signals from the main CPU. As such, we can only detect malware whose behavior has an impact on these signals. We currently extend the detection logic to the signal relative to the peripherals, as it seems more suited to detect malware inserted inside the peripherals themselves or Trojan inserted at the CPU level that need to communicate with the peripherals to execute their malicious payload.

This work to detect software attacks impacting the microarchitecture, or hardware Trojans, could be extended to reverse-engineering or forensic purposes at hardware level. Security counters can be stored internally or exported, in order to keep logs of past activities, and/or to characterize software/hardware interaction so as to identify the architecture used and the programs running on it, even if the supplier doesn't want to give that information. This would also enable subsequent failure diagnosis, or even remediation if partial reconfiguration is technically possible. On the industrial side, techniques already used in dependability (redundancy, majority voting, etc.) can be applied to provide effective countermeasures against failures, whether intentional such as fault injection, or not. Attacks could be designed specifically for these mechanisms, in order to consider future ways of protecting against them. Similarly, specific remediation capabilities for these devices could be devised, since critical systems cannot be interrupted, and may have to operate in degraded behavior.

References

1. Bourdon, M., Gimenez, P.F., Alata, E., Kaaniche, M., Migliore, V., Nicomette, V., Laarouchi, Y.: Hardware-performance-counters-based anomaly detection in massively deployed smart industrial devices. In: 2020 IEEE 19th International Symposium on Network Computing and Applications (NCA). pp. 1–8 (2020). https://doi.org/10.1109/NCA51143.2020.9306726
2. El-Bouazzati, M.: 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), https://cnrs.hal.science/tel-04612764
3. EnjoyDigital: Litescope - a small footprint and configurable embedded fpga logic analyzer (2015), https://github.com/enjoy-digital/litescope
4. Kermarrec, F., Bourdeauducq, S., Lann, J.C.L., Badier, H.: Litex: an open-source soc builder and library based on migen python dsl (2020), https://arxiv.org/abs/2005.02506
5. Mao, Y., Migliore, V., Nicomette, V.: Matana: A reconfigurable framework for runtime attack detection based on the analysis of microarchitectural signals. Applied Sciences **12**(3) (2022). https://doi.org/10.3390/app12031452, https://www.mdpi.com/2076-3417/12/3/1452
6. Mushtaq, M.: Software-based Detection and Mitigation of Microarchitectural Attacks on Intel's x86 Architecture. Theses, Université de Bretagne Sud (Sep 2019), https://theses.hal.science/tel-02988980
7. Polychronou, N.F., Thevenon, P.H., Puys, M., Beroulle, V.: Madman: Detection of software attacks targeting hardware vulnerabilities. In: 2021 24th Euromicro Conference on Digital System Design (DSD). pp. 355–362 (2021). https://doi.org/10.1109/DSD53832.2021.00060