ARMHEx: a framework for efficient DIFT in real-world SoCs

Muhammad Abdul Wahab, Pascal Cotret, Mounir Nasr Allah, Guillaume Hiet, Vianney Lapotre, Guy Gogniat

ARMHEx: a framework for efficient DIFT in real-world SoCs

Muhammad Abdul Wahab[α], Pascal Cotret[α], Mounir Nasr Allah[β], Guillaume Hiet[β]
Vianney Lapôtre[γ], Guy Gogniat[γ]
[α] IETR, SCEE research group, firstname.lastname@centralesupelec.fr
[β] INRIA, CIDRE research group, firstname.lastname@centralesupelec.fr
[γ] Univ. Bretagne-Sud, UMR CNRS 6285, Lab-STICC, firstname.lastname@univ-ubs.fr

***Abstract*—Security in embedded systems remains a major concern. Untrustworthy authorities use a wide range of software attacks. This demo introduces ARMHEx, a practical solution targeting DIFT (Dynamic Information Flow Tracking) implementations on ARM-based SoCs. DIFT is a solution that consists in tracking the dissemination of data inside the system and allows to enforce some security properties. In this demo, we show an implementation of ARMHEx on Xilinx Zynq SoC. Especially, we show how the required information for DIFT is recovered with the help of traces produced by CoreSight components, static analysis and instrumentation.**

I. Description

Software security remains a challenge for users and developers. Access control or cryptography can be used to limit access to confidential data or to enforce integrity. However, such techniques do not provide any guarantees once access is granted or data decrypted. DIFT is a promising technique that monitors information flows to ensure security properties. It consists in tagging each information container (e.g. variables, memory address, CPU registers, etc.), propagating tags when information flows take place and checking the value of tags according to a security policy. DIFT can successfully detect an important number of attacks varying from low-level threats (e.g. memory overflows) to high-level threats (such as data leakage).

Adding support in software for DIFT has an important execution runtime overhead [1]. To improve time overhead, Suh et al. [2] proposed the idea of using hardware acceleration. Since then, an important amount of work (such as [3]) has been done to provide acceleration of security features in FPGA. However, there is no related work that concentrates on providing these security features in real-world SoCs. There are two main reasons : the FPGA has limited or no visibility of executed software on CPU and the amount of time required to develop such a system. ARMHEx [4] overcomes previous limitations by proposing an efficient way of recovering required information for DIFT.

ARMHEx requires internal information about the program that is being executed on the CPU. A common solution in related work is to instrument software to recover required information for DIFT. However, instrumenting all instructions adds an important runtime overhead. Instead, ARMHEx proposes to combine static analysis and traces coming out of debug components to reduce number of instrumented instructions by 90% when compared to related work.

ARMHEx uses CoreSight components to trace the application that is being executed. The starting address of each basic block is known before execution. For instance, it can be obtained during compilation or by using a disassembler. By comparing the traces and basic blocks start addresses, ARMHEx knows which basic block is currently being executed. However, what happens inside each basic block remains unknown. Before executing the application, it is statically analyzed.

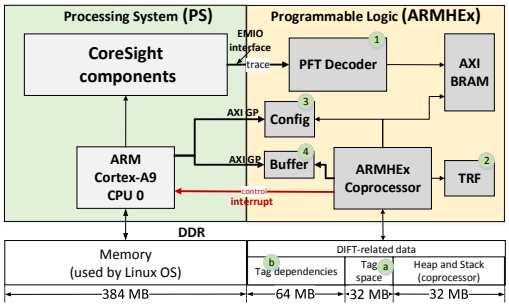

Fig. 1: Overall architecture of ARMHEx implemented on Zynq

In this demo, we show an implementation of overall architecture (Figure 1) and how it works to implement DIFT. A simple example ([4]) is considered and the operations done by ARMHEx coprocessor are shown and explained. The demo setup is shown in Figure 2.

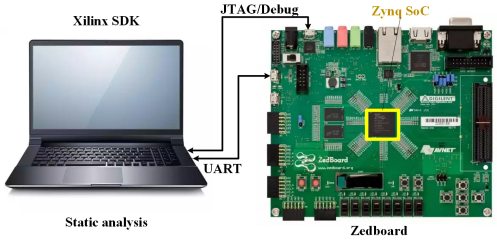

Fig. 2: Demo Setup

II. Acknowledgments

This work is done in the frame of HardBlare project which is funded by CominLabs and Brittany region.

References

[1] M. Dalton, H. Kannan, and C. Kozyrakis, "Raksha: A flexible information flow architecture for software security," *SIGARCH Comput. Archit. News*, vol. 35, no. 2, pp. 482–493, Jun. 2007.

[2] G. E. Suh, J. W. Lee, D. Zhang, and S. Devadas, "Secure program execution via dynamic information flow tracking," *SIGARCH Comput. Archit. News*, vol. 32, no. 5, pp. 85–96, Oct. 2004.

[3] J. Lee, I. Heo, Y. Lee, and Y. Paek, "Efficient security monitoring with the core debug interface in an embedded processor," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 22, no. 1, pp. 8:1–8:29, May 2016.

[4] M. Abdul Wahab, P. Cotret, M. Nasr Allah, G. Hiet, V. Lapotre, and G. Gogniat, "ARMHEx: A hardware extenstion for DIFT on ARM-based SoCs," in *27th International Conference on Field-Programmable Logic and Applications (FPL 2017)*, Sep. 2017, in press.