HardBlare: an efficient hardware-assisted DIFC for non-modified embedded processors

Pascal Cotret, Guillaume Hiet, Guy Gogniat, Vianney Lapotre

HardBlare: an efficient hardware-assisted DIFC for non-modified embedded processors

Pascal Cotret[α], Guillaume Hiet[β], Guy Gogniat[γ] and Vianney Lapôtre[γ]

[α] SCEE/IETR, CentraleSupélec, Cesson-Sévigné - FRANCE
[β] CIDRE/INRIA, CentraleSupélec, Cesson-Sévigné - FRANCE
[γ] Lab-STICC, University of South Brittany, Lorient - FRANCE

Information Flow Control is a security mechanisms that provides security guarantees about information propagation. Other security mechanisms such as access control or cryptography can be used to limit the dissemination of confidential information and the modification of high integrity contents. However, they do not enforce end-to-end properties. They cannot control the dissemination of information once file access is allowed or the data is decrypted. In this context, HardBlare proposes a software/hardware codesign methodology to ensure that security properties are preserved all allong the execution of the system but also during files storage. The general context of HardBlare is to address Dynamic Information Flow Control (DIFC) that generally consists in attaching marks (also known as tags) to denote the type of information that are saved or generated within the system.

DIFC can be achieved at the software level. This coarse-grained monitoring technique attaches labels to information containers such as files, memory pages or IPCs. However, its adoption is limited for two main reasons:

- It is incompatible with existing applications and hardware drivers.
- Main drawback, it implies large time overheads (at least, $\times 3$).

That is the reason why some approaches rely on a hardware-assisted solution: hardware DIFC consists in modifying existing hardware to accelerate tags propagation and computation. Hardware mechanisms could also be used to protect tags in volatile memory. This hardware mechanism can be used directly to monitor the information flow inside applications that have been compiled into machine code. Some recent works such as [1, 2, 3, 4, 5] have already proposed hardware mechanisms for DIFC. Nevertheless, all these works rely on modifications of the processor running the main application. Kannan et al. [3] proposes a solution entirely implemented on a FPGA with a detached coprocessor aiming to speed up the DIFC controls but they do not take care of the coprocessor security at the hardware level and do not explore all possibilities that can be performed with FPGAs (partial reconfiguration, multicores use case and so on).

On a practical point of view, security solutions based on hardware and software modifications are hardly adopted. This is for a large part due to the cost of these hardware modifications but also to the cost induced by the redevelopment of the whole software stack to be adapted to this specific hardware. HardBlare tackles these issues by combining a non-modified processor core and the use standard existing OS (like Linux). This work presents an analysis of the main hardware-assisted DIFC approaches and the improvements brought by HardBlare. Furthermore, it presents new directions in the context of DIFC and first results perspectives for HardBlare mechanisms implemented on a Zynq SoC combining an ARM Cortex-A9 (ARMv7 architecture) with FPGA fabric.

Contact: pascal.cotret@centralesupelec.fr, +33(0)2.99.84.45.77

References

[1] S. Chiricescu, A. DeHon, D. Demange, S. Iyer, A. Kliger, G. Morrisett, B.C. Pierce, H. Reubenstein, J.M. Smith, G.T. Sullivan, A. Thomas, J. Tov, C.M. White, and D. Wittenberg. Safe: A clean-slate architecture for secure systems. In *Technologies for Homeland Security (HST), 2013 IEEE International Conference on*, pages 570–576, Nov 2013.

[2] Michael Dalton, Hari Kannan, and Christos Kozyrakis. Raksha: a flexible information flow architecture for software security. In *34th International Symposium on Computer Architecture (ISCA 2007), June 9-13, 2007, San Diego, California, USA* [2], pages 482–493.

[3] H. Kannan, M. Dalton, and C. Kozyrakis. Decoupling dynamic information flow tracking with a dedicated coprocessor. In *Dependable Systems Networks, 2009. DSN '09. IEEE/IFIP International Conference on*, pages 105–114, June 2009.

[4] G. Edward Suh, Jae W. Lee, David Zhang, and Srinivas Devadas. Secure program execution via dynamic information flow tracking. *SIGARCH Comput. Archit. News*, 32(5):85–96, October 2004.

[5] Guru Venkataramani, Ioannis Doudalis, Yan Solihin, and Milos Prvulovic. Flexitaint: A programmable accelerator for dynamic taint propagation. In *14th International Conference on High-Performance Computer Architecture (HPCA-14 2008), 16-20 February 2008, Salt Lake City, UT, USA* [5], pages 173–184.