PhD Position: Deployment of secured virtual machines

The MOCS team at Lab-STICC ENSTA-Bretagne is searching for a young, motivated and skilled PhD researcher with a strong background in computer engineering.

Position: PhD student
Duration: 36 months *starting date:* September/October 2018
Requirements: Master + European citizen
Where: MOCS team at Lab-STICC ENSTA-Bretagne, Brest
Scientific advisors: Loïc LAGADEC
Contact info: loic.lagadec@ensta-bretagne.fr

About Lab-STICC ENSTA-Bretagne

ENSTA Bretagne was founded in 1971 and is a multidisciplinary engineering institute under the auspices of the French Defence Ministry (DGA). ENSTA Bretagne has established itself in the field of IT oriented research, thought its laboratory Lab-STICC (UMR 6285), standing for ”Information and Communication Science and Technology Laboratory. Lab-STICC is a French National Centre for Scientific Research (CNRS) mixed unit shared with two universities and two other engineering institutes.

Lab-STICC’s focus can be summed up in the following motto: COMMUNICATE and DECIDE « from sensor to knowledge » standing for bringing solutions for physical layers at radio- frequency level, designing data transmission and management systems based on advances in both algorithmic and micro-electronic fields, and analysing information to deliver knowledge to final users.

Lab-STICC ENSTA Bretagne has been developing several tools that perform system level design and verification (OBP, http://www.obpcdl.org/), high-level synthesis (Morphose), and low-level reconfigurable architecture modelling & design kit (Madeo), that are currently being retargeted to security concerns.

Context

Embedded systems are, today, at the hearth of most complex systems and the tendency is towards an increased reliance on the use of dynamically “programmable” software and architectures even in security-critical systems. In this context, the French government emphasizes the need to acquire a nation-wide capability in the field of cyber-security enabling the protection against cyber-attacks of vital infrastructures and systems.

Virtual machines are now deployed in a massive way. They allow application portability between different hardware platforms, durability, and full use of available resources (secure sharing of resources, load balancing, migration, etc.). In doing so, virtual machines are taking an increasingly important place in current execution stacks (mobile phones, web browsers, large critical infrastructure servers: stock exchanges, banks, insurance, hospitals, etc., and embedded systems [1] like those produced by major players in the defence industry).

This situation leads to a high exposure to cyber risks. Indeed, a service built above these virtual machines is neither safe nor secured without the control of the VM implementation. However, virtual machine know-how and technology are concentrated in the hands of a few American companies: Google (javaScript - Android), Microsoft (DotNet), Oracle (Java).

Research Topic

The study will start by analysing the existing open-source virtual machines: Pypy (Python) [A], Pharo (Smalltalk) [B], LLVM (multilingual virtual machine) [C], Truffle / Graal (multilingual virtual machine) [D] / E]. This aims at identifying the strengths and weaknesses of all different alternatives, particularly in a cyber security context.

Projects that have highlighted the limitations of certain approaches will be reported. As examples, the VMKit project [2, F] initiated by Gaël Thomas and today discontinued, has identified the limits of LLVM in a context of implementation of virtual machines for dynamic languages, whereas if Apple (virtual machine Javascript Webkit for Safari) has successfully taken benefit from LLVM, it has also led to a much larger work than expected [3] to the point of going back on this decision and to implement a dedicated back-end [4].

The refactoring of virtual machine will be at the heart of the approach (bootstrap, image maker, componentization, etc.) as well as its functional characteristics (adding/removing on the fly compilation, introspection, etc.).

We will then focus on the transposition to virtual machines of the microkernel approach [5], which leads to isolate a minimal core from a set of possible extensions.

One hot topic will be reducing memory footprint, which is critical for embedded systems and IoT. A second target will be to extend the VM with some advanced capabilities, depending on the context: memory manager, file manager, hot debug, etc. to balance the flexibility and maintainability versus security tradeoff.

The ultimate goal is to enable a "product line" [6] approach for the design of specific and secure virtual machines, targeting platforms ranging from computer servers to systems-on-a-chip.

References:

Internal References

[E1] C. Teodorov, L. Lagadec: *Model-driven physical-design automation for FPGAs: fast prototyping and legacy reuse*. In Software, Practice and Experience. 44(4): 455-482 (2014)

[E2] Y. Corre, J-Ph. Diguet, D. Heller, D. Blouin, L. Lagadec: *TBES: Template-Based Exploration and Synthesis of Heterogeneous Multiprocessor Architectures on FPGA*. ACM Trans. Embedded Comput. Syst. 15(1): 9:1-9:27 (2016)

[E3] L. Lagadec, C. Teodorov, J-Ch. Le Lann, D. Picard, E. Fabiani: *Model-driven toolset for embedded reconfigurable cores: Flexible prototyping and software-like debugging.* In Science of Computer Programing. 96: 156-174 (2014)

[E4] M. Ben Hammouda, Ph. Coussy, L. Lagadec. A Unified Design Flow to Automatically Generate On-Chip Monitors During High-Level Synthesis of Hardware Accelerators. In IEEE Transactions on CAD of Integrated Circuits and Systems. Vol 36, no 3, 2017, pages 384-397. doi : 10.1109/TCAD.2016.2587278

[E5] C. Feron, V. Lapotre, L. Lagadec: *PAnTHErS: A Prototyping and Analysis Tool for Homomorphic Encryption Schemes.* SECRYPT2017: 359-366

[E6] G. Polito, C. Teruel, S. Ducasse, L. Fabresse: *Scoped Extension Methods in Dynamically-Typed Languages*. Programming Journal 2(1): 1 (2018)

[E7] G. Polito, S. Ducasse, L. Fabresse, N. Bouraqadi, B. Van Ryseghem: *Bootstrapping reflective systems: The case of Pharo.* In Science of Computer Programing 96: 141-155 (2014)

[E8] C. Béra, E. Miranda, M. Denker, S. Ducasse: *Practical Validation of Bytecode to Bytecode JIT Compiler Dynamic Deoptimization. In* Journal of Object Technology 15(2): 1:1-26 (2016)

[E9] G. Polito, L. Fabresse, N. Bouraqadi, S. Ducasse: *Run-Fail-Grow: Creating Tailored Object-Oriented Runtimes.* Journal of Object Technology 16(3): 2:1-36(2017)

External References

[1] A. L. Sartor, A. F. Lorenzon, A. C.S. Beck, *The impact of Virtual Machines on Embedded Systems*, in 39th IEEE Computer Software and Applications Conference (COMPSAC), 2015

[2] N. Geoffray, G. Thomas, J. Lawall, G. Muller and B. Folliot. *VMKit: a substrate for managed runtime environments*. In Proceedings of the international conference on Virtual Execution Environments, VEE'10, pages 51-62. 2010.

[3] Introducing the WebKit FTL JIT. https://webkit.org/blog/3362/introducing-the-webkit-ftl-jit/

[4] Introducing the B3 JIT Compiler. https://webkit.org/blog/5852/introducing-the-b3-jit-compiler/

[5] K. Wang, Y. Lin, S. M. Blackburn, M. Norrish, and A. L. Hosking*, Draining the Swamp: Micro Virtual Machines as Solid Foundation for Language Development,* in Summit on Advances in Programming Languages 2015. doi : 10.4230/LIPIcs.SNAPL.2015.321

[6] M. Haupt, S. Marr, R. Hirschfeld. *CSOM/PL A Virtual Machine Product Line*. In Journal of Object Technology, vol. 10, no. 12, 2011, pages 1–30. doi:10.5381/jot.2011.10.1.a12

Environments

[A] https://pypy.org
[B] https://pharo.org
[C] http://llvm.org
[D] https://github.com/oracle/graal/tree/master/truffle
[E] https://github.com/oracle/graal
[F] VMKit: a substrate for virtual machines. Official Website. https://vmkit.llvm.org/
[G] ESUG European Smalltalk User Group. http://www.esug.org