Research Internship 2021-2022 Topic

Measures against speculative attacks in a certified optimizing compiler

Supervisors: David Monniaux and Sylvain Boulmé at Verimag[*]

October 11, 2022

CompCert[1] is a compiler for the C programming language for the assembly languages of several processor architectures. In contrast to compilers such as Visual C++, GCC, or LLVM, its compilation phases are proved mathematically correct, and thus the compiled program always matches the source program: the formal correctness of CompCert states that if the compiler succeeds to produce an executable, then the *observable behaviors* of the executable are also *observable* on the source program [1, 2]. Other compilers may contain bugs that in some cases result in incorrect code being generated. The possibility of compilation bug cannot be tolerated in certain applications with high safety requirements, and then costly solutions such as disabling all optimizations are used to get assembly code that is close to the source. In contrast, CompCert, despite not optimizing as well as gcc -O3 or clang -O3, allows using optimizations safely [3, 4].

Speculative execution is a feature of most current high-performance processors whereby the processor starts executing instructions even though it is still unsure that these could be legally executed (for instance, it is unsure that branches actually lead into that execution). If it turns out that these instructions should not have been executed, architectural effects (e.g., writes to registers, writes to memory, etc.) are undone. However, microarchitectural effects (e.g., the use of CPU units and busses, the loading of cache lines) are not undone.

In January 2018, the *Spectre* and *Meltdown* vulnerabilities took the world by storm, and inspired further attack schemes. These vulnerabilities exploit side channel attacks on behaviors attained through speculative execution; for instance, they may exploit a side channel through cache timing attacks.

Various mitigation measures have been implemented so far, including some in the gcc and LLVM compilers.

The goal of this internship is to study the Spectre and variant vulnerabilities, as well as the associated compile-time mitigations, and design and implement some in CompCert.

[*] http://www-verimag.imag.fr/, mail at *firstname.lastname*@univ-grenoble-alpes.fr

[1] https://compcert.org/