Reliable Fast Stack Unwinding

Francesco Zappa Nardelli
francesco.zappa_nardelli@inria.fr
Parkas Project-Team, INRIA Paris & ENS, France

Stack unwinding is a key operation relied upon not only by debuggers, but also by most program analysis tools and by the runtime of some high-level programming languages to implement constructs as C++ exceptions. The general technique to implement stack unwinding currently requires reading out-of-band DWARF debugging information, decoding them, and interpreting them on a Turing-complete stack-based machine that can read any machine register and dereference arbitrary memory. As a consequence, stack unwinding is a slow and not as reliable as such key operation should be.

In this internship we will design and implement techniques to:

• validate the correctness of debug information against the binary code. This will increase assurance in their correctness and is a first step to turn stack unwinding into a reliable process;

• study techniques to compile ahead-of-time the out-of-band debug information. Unwinding thus will not need to interpret on the fly by the unwinder, but it will be possible to directly jump to ad-hoc assembly code.

These goals leverage formal techniques, including symbolic execution of assembly code and partial evaluation. As customary in system research, additional complexity comes from the need to support complete standards, without the possibility of focusing on toy-languages. We will implement our techniques on state of the art tools (e.g. libunwind for the second goal) and evaluate experimentally their benefits on large and challenging corpus of code as the Glibc and the Linux kernel.