technicolor

Technicolor R&D France Snc
975 av des Champs Blancs – CS 17616
35576 Cesson-Sévigné Cedex - France

tél. + 33 (0)2 99 27 30 00
fax + 33 (0)2 99 27 30 01

Internship proposal for 2015
Proposal ref:
DPL_2015_SP_009

Title of internship : DPL_2015_SP_009 Design and Development of a LLVM Module for Code Renewability

Summary of the internship (for Technicolor internship offer web page).

	The goal of this internship is to modify a compiler framework in order to be able to produce a different binary code at each new compilation of the same source code. The goal is to increase the effort of reverse engineering required once a given binary code has been hacked Skills : compilers knowledge and desire to learn software protection Keywords : Software protection, compiler, LLVM, C++

Detailed description

Context
In order to break an application, an attacker generally needs a deep knowledge of the application's code. This acquisition requires a lot of efforts. However, once an attacker finds a flaw and exploits it, a simple correction will not modify deeply the binary code. Therefore, the effort needed to find another flaw will be importantly reduced thanks to the acquired knowledge. To avoid this, the developer needs to generate a new binary code that looks very different from the first one. Some studies* have already been made to generate different binaries compiling the same source code. However, proposed solutions were mainly aimed at protecting cryptographic algorithms from side channel attacks and are not applicable in a broader context. *Damien Couroussé, Bruno Robisson, Jean-Louis Lanet, Thierno Barry, Hassan Noura, Philippe Jaillon and Philippe Lalevée.(august 2014) COGITO : Code Polymorphism to Secure Devices

Apply at: stage.rennes@technicolor.com

Objective

The goal of this internship is to design and developed a module in the LLVM framework to generate a different binary at each compilation of the same source code.
The source code used will be written in C and C++.
An important aspect of this intern will be to manage the performance overhead due to the introduction of randomness that will break some optimization. Usually the compiler can produce a lot of different binaries for a same source code and the choice is driven by performance. Here the goal is to define the degree of randomness we can accept on different parameters to generate different code with acceptable execution time.

Task description

The intern will start by a study of the state of the art. Then he will have to find some different ways to insert some randomness in the compilation chain. The intern will have to design and develop his solution using LLVM and to measure the performance overhead due to his transformation. Indeed the introduction of randomness will break some optimisation techniques and we will have to find a way to limit the overhead.
The design and the development will be modular. Then it will be possible to start with one parameter like the scheduler and when the implementation using the “random “scheduling will work the intern will be able to work on a second parameter and son on.

Working environment

LLVM
C, C++
Compiler
Assembler
Linux