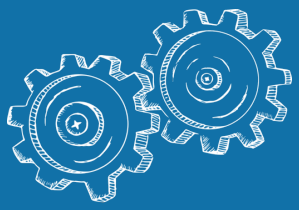

Implementing a LLVM Compiler Mitigation Pass

Advisor: **Andreas Kogler and Claudio Canella**

Motivation

Both architectural (ROP [1]) and microarchitectural attacks (LVI [2]) pose a large problem for the overall security of the system.

For either attack scenario, reducing the attack surface by hand is complicated and not feasible for large scale projects. Therefore, mitigations against such attacks are often implemented inside compilers to automatically prevent them.

In this thesis, we want you to implement and evaluate an additional compiler pass inside the LLVM compiler infrastructure [3] to mitigate a possible attack. You will learn about different attacks and how to modify and extend a state of the art compiler to prevent them.

Goals and Tasks

- Get familiar with the attack surface
- Implement the mitigation inside the compiler
- Evaluate the performance impact of the mitigation

Literature

- H. Shacham
  The geometry of innocent flesh on the bone: Return-into-libc without function calls (on the x86)
- J. Van Bulck et al.
  LVI: Hijacking Transient Execution through Microarchitectural Load Value Injection
- LLVM
  The LLVM Compiler Infrastructure
  https://llvm.org

Courses & Deliverables

- ☑ **Introduction to Scientific Working**
  Short report on background
  Short presentation
- ☑ **Bachelor Project**
  Project code and documentation
- ☑ **Bachelor's Thesis**
  Project code
  Thesis
  Final presentation

Recommended if you're studying

☑ CS ☑ ICE ☑ SEM

Prerequisites

- Programming (C/C++, Assembly)

Advisor / Contact

andreas.kogler@iaik.tugraz.at;claudio.canella@iaik.tugraz.at