Motivation

- Control-flow and data-oriented run-time attacks alter program behaviour
- Control- and Data-flow Integrity detect invalid control- or data-flows caused by attacks

ARM Pointer Authentication

- ARMv8.3-A provides instructions for creating and validating Pointer Authentication Codes
- MAC using tweakable QARMA cipher
- Calculated from pointer value, tweak and process-specific key

Context based problems

- Tweak derived from pointer usage context
  - Must be uncontrollable by attacker and sufficiently unique
- GCC backward-edge CFI uses only stack pointer value as context for return address

Proposed Solutions

- PA based CFI and DFI
- Back- and forward-edge CFI
- DFI for protecting data pointers
- Diverse context to prevent rollback/swapping
- Run-time type safety via type-based context

Implementation

- LLVM instrumentation additions to 64bit ARM backend, optimization passes, and clang
- Kernel support for PA data key management
- Loader support for PACing PLT and GOT

void f1() {
    ...
    return;
}

void do_stuff() {
    func();
    if (a) func();
}

PA instrumented code vulnerable to return address substitution