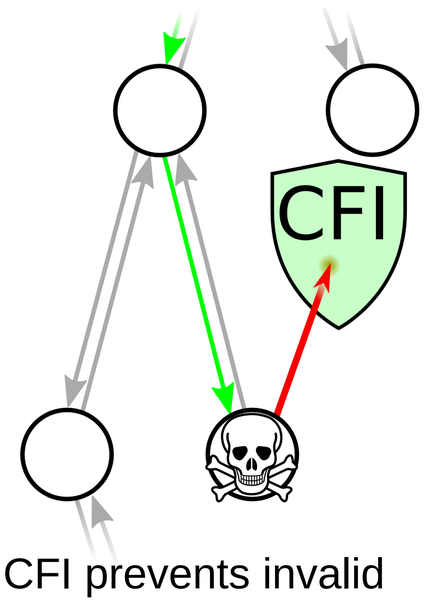
Platform Security
Secure Systems Group, Aalto University

Hans Liljestrand, Thomas Nyman, Kui Wang, Carlos Chinea Perez, Jan-Erik Ekberg, N. Asokan

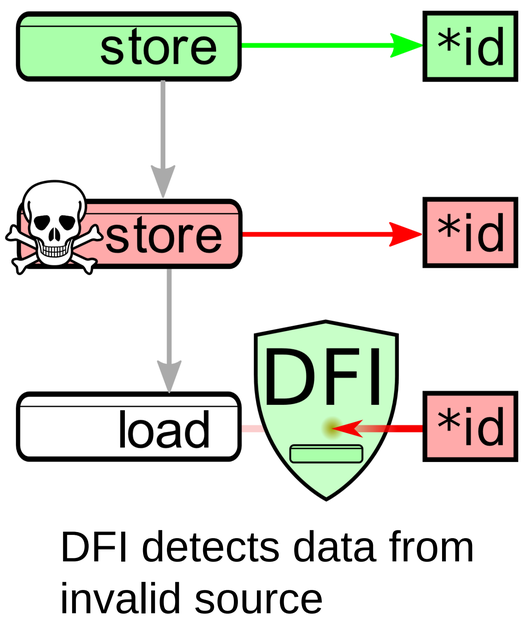
Code- and Data-flow Integrity using ARM Pointer Authentication

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



CFI prevents invalid control flow transitions

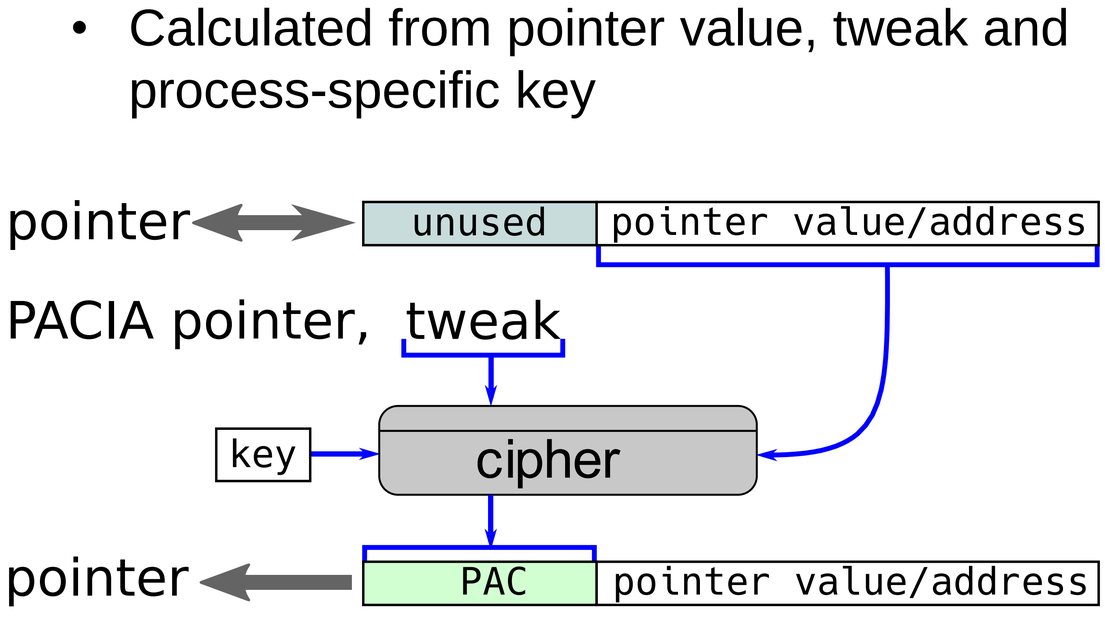


DFI detects data from invalid source

- Hardware Pointer Authentication can ease implementation and improve performance

ARM Pointer Authentication

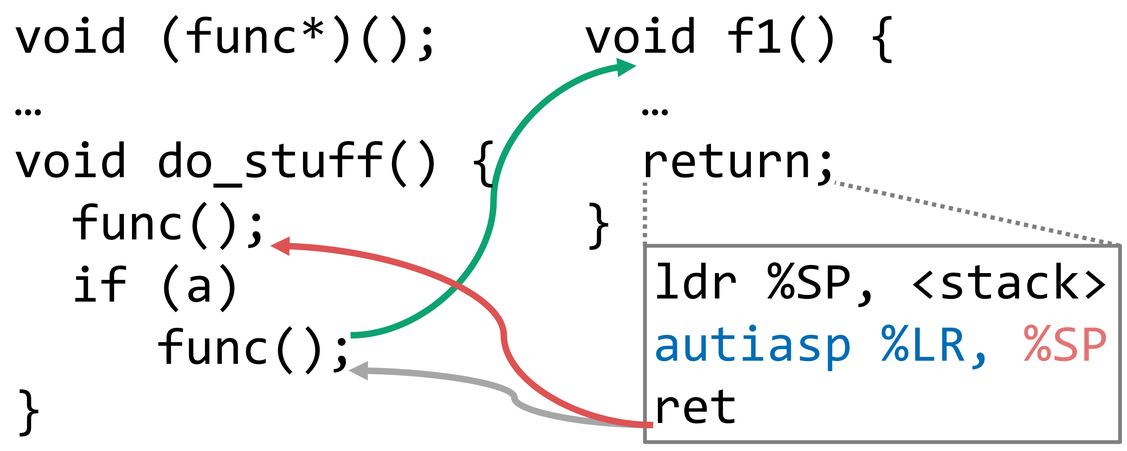
- ARMv8.3-A provides instructions for creating and validating Pointer Authentication Codes
  - MAC using tweakable QARMA cipher
  - Calculated from pointer value, tweak and process-specific key



- ARM added basic support for PA in GCC 7

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



PA instrumented code vulnerable to return address substitution

- Challenging to choose diverse usage context
  - Must be known on creation and use
  - Not always known compile-time
  - Run-time context requires safe storage

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

Aalto University

HUAWEI