Control Flow Integrity of Dynamic Programming Using an FPGA Board

sdmay21-10

Gregory Wendt - Meeting Scribe
Cole Schumacher - Meeting Facilitator
Nickolas Mitchell - Chief Engineer (FPGA)
Sam Henley -Chief Engineer (Software)
Maxwell Wrangler - Test Engineer
Tristan Duyvejonck - Report Manager

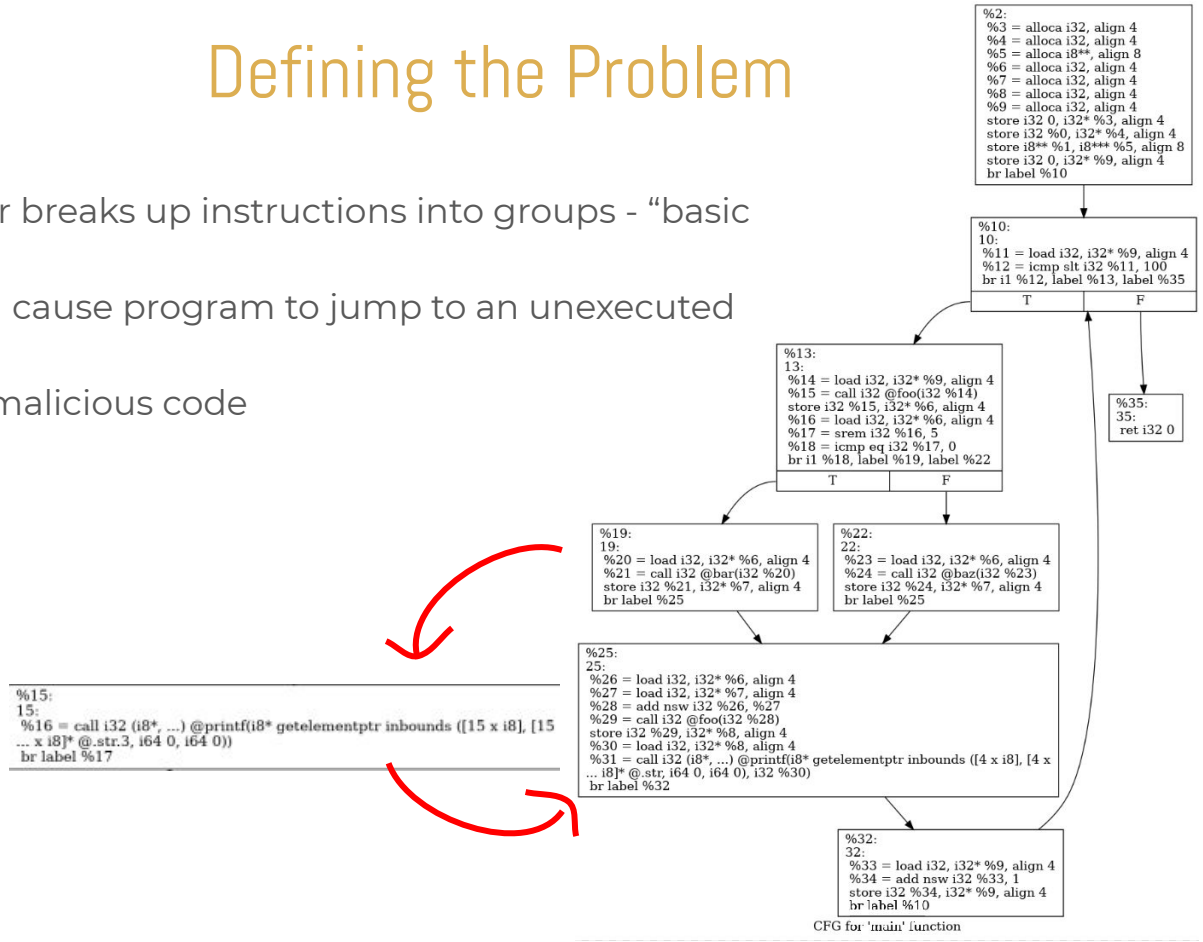
Contact:
sdmay21-10@iastate.edu

Client:
Dr. Akhilesh Tyagi

Faculty Advisors:
Zelong Li
Ananda Biswas

Defining the Problem

- The compiler breaks up instructions into groups - “basic blocks”
- Attacker can cause program to jump to an unexecuted basic block
  - Inject malicious code

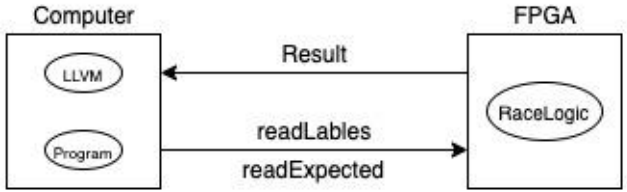


Project Goal

Detect breaches in control-flow integrity using dynamic programming on an FPGA board

We will do this by:

- Creating race logic in a higher level language to test.
- Integrating race logic into hardware (FPGA).
- Set up communicate between computer and FPGA.



Current Technical Challenges - FPGA

Test VHDL Code

- Changes from software solution include:
  - 8-bit inputs
  - Non-Changeable string length
- Test using communication between computer and FPGA

Communication Between Computer and FPGA

- FPGA - USB using UART
- Computer - PuTTY or Visual Studio

Current Technical Challenges - LLVM

LLVM: Compiler and toolchain

Convert source code to Intermediate Representation (IR)

- Portable, high-level assembly language

Passes: transformations or optimizations to a program

- 1st Pass: read program and get sequence of basic blocks
- 2nd Pass: inject function at the end of each basic block, generate sequence of basic blocks encountered at runtime

Issues:

- How to relate block identifiers between passes
- How to get data to FPGA

Standards - IEEE

- IEEE 1008-1987: IEEE Standard for Software Unit Testing
- IEEE 1500-2005: IEEE Standard Testability Method for Embedded Core-based Integrated Circuits
- IEEE 15288-2004: Systems and Software Engineering System Life Cycle Processes
- IEEE 1220-2005: IEEE Standard for Application and Management of the Systems Engineering Process

Requirements and Constraints

Engineering Requirements

- Create an algorithm to match two sets of strings (race logic).
- One string string generated based on control flow, and compared to a defined string.
- Use a 2-D grid of which the size is determined by the number of basic blocks.
- Algorithm must perform in O(n log(n)) runtime.

Engineering Constraints

- Use dynamic programming to determine if CFI was lost.
- Use a FPGA board to test the solution.
- Total project cost should not exceed $100.

Questions?

Contact:
sdmay21-10@iastate.edu