A Processor + FPGA based Platform for Control Flow Integrity Enforcement

Student: Anirudh Iyengar, Advisor: Swaroop Ghosh & Trent Jaeger

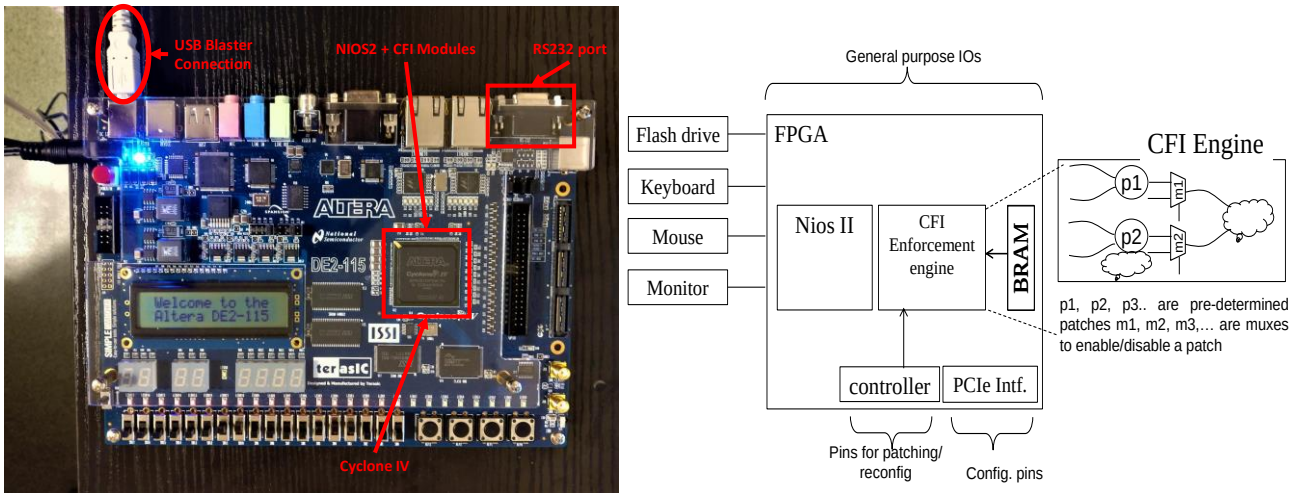
1. Description of Research

The objective of this research is to maintain integrity against control flow hijacking using reconfigurable platform. Existing security solutions require instrumentation of code that makes the software inflexible. The additional code for runtime enforcement of integrity validation degrades performance. The secure hardware platforms such as, ARM Trustzone require significant design overhead, code changes, and, affect performance due to restricted access policies. Additionally, the hardware is not amenable to patching to address evolving threats. This project will employ heterogeneous reconfigurable computing platforms created using FPGA (Field Programmable Gate Array) to enforce security and circumvent performance impact. In heterogeneous platform, the FPGA will be configured to validate the processor code execution to a predefined Control Flow Graph (CFG) through Control Flow Integrity (CFI) enforcement engine. This approach will eliminate the instrumentation of code, keeping them flexible without sacrificing performance. The reconfigurability of FPGA will offer provisions to patch the hardware with latest updates and capabilities in sync with the evolving threat space and enable dynamic tradeoff between performance and security. We will illustrate the effectiveness of this technique against standard benchmarks such as CoreMark, Dhrystone etc.

2. Description of Demonstration

The FPGA will be programmed to emulate two modules (Fig. 1 (b)): (i) the soft-core processor (Nios2 in our demonstration); and, (ii) CFI enforcement engine. We will first demonstrate the base operating mode of the Nios2 processor and the CFI module as shown in Fig. 1(a). Following which, the system will be benchmarked using standard CPU benchmarks to obtain the base performance values. The next step will be to expose this system to a tainted set of benchmarks, in-order to analyze the effectiveness of the proposed CFI validation engine. Patching of the enforcement engine will be also demonstrated by using pre-designed patches and unmasking them to add new features or enable new policies (Fig. 1(b)). The performance benefits of the proposed technique will be compared with standard processor with software-based CFI enforcement that we have implemented in past [1] [2].

References: [1] Ge, Xinyang, et al. "Fine-Grained Control-Flow Integrity for Kernel Software." 2016 IEEE European Symposium on Security and Privacy (EuroS&P). IEEE, 2016. [2] Xinyang Ge, Weidong Cui, Trent Jaeger. GRIFFIN: Guarding Control Flows Using Intel Processor Trace. ASPLOS, April 2017



(a) (b)

Fig. 1 (a) Test setup. We employ an Altera DE2-115 board with Cyclone IV configured to emulate a Nios2 processor with CFI modules. A computer is connected to tally and interpret results; and, (b) conceptual picture of the proposed approach. The CFI engine will monitor the processor traffic and validate CFI. Pre-defined masked patches will be incorporated in the enforcement engine which will be unmasked to unveil new features.