Security Techniques in Cyber Physical Systems

Christopher Zimmer
CSC 714

Introduction

Typical cyber-physical systems interact and control devices that we take for granted in our daily lives. As exposure to these systems increase the need to secure them from attackers does as well. Potential attacks could affect such areas as the brakes in our cars, the elevators we ride in, potentially even the power grids we use on a daily basis. The goal of this project is to explore methods of securing these systems and introduce the concept of intrusion detection to this domain of technology. Current work in security is primarily geared toward securing general-purpose systems from attack. This is partially due to the visible prevalence of these systems and their exposures to open networks. Similarly cyber-physical systems designs more frequently now include network interface devices to enable communication for these systems. The serious threat this creates is a more conspicuous interface for an attacker to utilize.

Proposal

The primary goal of this work is to introduce new security features that can be developed in the real-time feature space. In this work I'm going to continue work in applying security techniques to real-time systems. I'm going to expand current work in utilizing temporal analysis in detecting security based timing anomalies in real-time systems. In this work we are going to develop a new model for a periodic scheduler to maintain timing state inside of a real-time simulator (SimpleScalar). This work will require modifying the current simple scalar and scheduler implementations to support periodic wake-up, back-edge counting, and internal task checking from the scheduler utilizing pc values. This work will also require breaking up the timing analysis of a program in to different states that can be mapped within the scheduler without creating an excess amount of overhead.

Proposed Work Outline

Week 1

- A full literature search of hardware and software security approaches more specifically in the realm of real-time systems, but because the area is relatively novel, a much broader search will be employed as well.

Week 2

- Analyze simulator and identify components within the simulator that may exist to aid in this work. i.e. the Existence of timer interrupts, back-edge detection mechanisms.

Weeks 3 and 4

- Modify Real-Time Scheduler in system to support low overhead periodic checks targeted solely towards security. Analyze other interrupts that may occur and analyze the potential

Week 5

- Fit Timing Tree's into scheduler to enable PC based lookups and verification regardless of program input.
- Test against CLAB Benchmarks

Week 6

- Finish collecting experimental benchmarks.
- Write up results into paper for submission

Some of the major issues that will have to be handled during the implementation of this project

- Partitioning program into regions so that WCET may be accurately checked against a PC
- Modifying Simulator to Support maintaining backedge counts between interrupts or another mechanism to employing this.
- Evaluating different timing regions within the tasks and determining the best way in the security task to check these timing regions.
- Handling preemptions and other side effects of a real-time task set.

The URL for this work can be found at
http://www4.ncsu.edu/~cjzimme2/

Reference:

1. "Bounding Worst-Case Instruction Cache Performance" by R. Arnold, F. Mueller, D. B. Whalley and M. Harmon in IEEE Real-Time Systems Symposium, Dec 1994, pages 172-181 http://moss.csc.ncsu.edu/~mueller/publications.html #19

2. "Bounding Worst-Case Data Cache Behavior by Analytically Deriving Cache Reference Patterns" by H. Ramaprasad and F. Mueller in Real-Time and Embedded Technology and Applications Symposium, March 2005, pages 148-157.(slides). http://moss.csc.ncsu.edu/~mueller/ftp/pub/mueller/papers/rtas05dcache.pdf

3. J. Renau, B. Fraguela, J. Tuck, W. Liu, M. Prvulovic, L. Ceze, S. Sarangi, P. Sack, K. Strauss, and P. Montesinos. SESC Simulator, Jan. 2005, http://sesc.sourceforge.net.