Matthew Hicks

mdhicks@gmail.com
www.ImpedimentToProgress.com
github.com/impedimentToProgress

60 Garnet Rock Ln
Carlisle, MA 01741
(217) 766-4294

RESEARCH INTERESTS

Broadly, I am interested in systems, security, and architecture. My current research spans hardware and embedded system security, intermittently powered systems, and approximate computing. Previous research projects include, crafting and detecting malicious insertions into hardware and hardware support for real-time and embedded systems.

EDUCATION

Doctorate, Computer Science **May 2013**
University of Illinois Urbana-Champaign
Hybrid approaches for overcoming processor imperfections

Master of Science, Computer Science **August 2008**
University of Illinois Urbana-Champaign
Real-time Systems

Bachelor of Science, Computer Science **May 2006**
University of Central Florida
Mathematics Minor
Honors College Graduate

RECOGNITION

2016 IEEE Symposium on Security and Privacy Distinguished Paper Award
2016 Pwnie Awards Most Innovative Research Award finalist

CONFERENCE PUBLICATIONS

Clank: Architectural Support for Intermittent Computation. Matthew Hicks. International Symposium on Computer Architecture (**ISCA**). June 2017.

Intermittent Computation without Hardware Support or Programmer Intervention. Joel Van Der Woude and Matthew Hicks. USENIX Symposium on Operating Systems Design and Implementation (**OSDI**). November 2016.

A2: Analog Malicious Hardware. Kaiyuan Yang, Matthew Hicks, Qing Dong, Todd Austin, and Dennis Sylvester. IEEE Symposium on Security and Privacy (**Oakland**). May 2016.
Distinguished Paper award.
Pwnie Most Innovative Research Award finalist.

ANVIL: Software-Based Protection Against Next-Generation Rowhammer Attacks. Zelalem Birhanu Aweke, Salessawi Ferede Yitbarek, Rui Qiao, Reetuparna Das, Matthew Hicks, Yossi Oren, and Todd Austin. Symposium

on Architectural Support for Programming Languages and Operating Systems (**ASPLOS**). March 2016.

Probable Cause: The Deanonymizing Effects of Approximate DRAM. Amir Rahmati, Matthew Hicks, Daniel E. Holcomb, and Kevin Fu. International Symposium on Computer Architecture (**ISCA**). June 2015.

SPECS: A lightweight runtime mechanism for protecting software from security-critical processor bugs. Matthew Hicks, Cynthia Sturton, Samuel T. King, and Jonathan M. Smith. Symposium on Architectural Support for Programming Languages and Operating Systems (**ASPLOS**). March 2015.

Defeating UCI: Building Stealthy and Malicious Hardware. Cynthia Sturton, Matthew Hicks, David Wagner, and Samuel T. King. IEEE Symposium on Security and Privacy (**Oakland**). May 2011.

Overcoming an Untrusted Computing Base: Detecting and Removing Malicious Hardware Automatically. Matthew Hicks, Murph Finnicum, Samuel T. King, Milo M. K. Martin, and Jonathan M. Smith. IEEE Symposium on Security and Privacy (**Oakland**). May 2010.

Capo: a Software-Hardware Interface for Practical Deterministic Multiprocessor Replay. Pablo Montesinos, Matthew Hicks, Samuel T. King, and Josep Torrellas. Symposium on Architectural Support for Programming Languages and Operating Systems (**ASPLOS**). March 2009.

OTHER PUBLICATIONS

SNIFFER: A High-Accuracy Malware Detector for Enterprise-based Systems. Evan Chavis, Harrison Davis, Yijun Hou, Matthew Hicks, Salessawi Ferede Yitbarek, Todd Austin, and Valeria Bertacco. International Verification and Security Workshop. July 2017.

Approximate Flash Storage: A Feasibility Study. Amir Rahmati, Matthew Hicks, and Atul Prakash. Workshop on Approximate Computing Across the System Stack. March 2016.

Refreshing Thoughts on DRAM: Power Saving vs. Data Integrity. Amir Rahmati, Matthew Hicks, Daniel E. Holcomb, and Kevin Fu. Workshop on Approximate Computing Across the System Stack. March 2014.

Practical systems for overcoming processor imperfections. Matthew Hicks. University of Illinois Urbana-Champaign, PhD Dissertation. May 2013.

Overcoming an Untrusted Computing Base: Detecting and Removing Malicious Hardware Automatically. Matthew Hicks, Murph Finnicum, Samuel T. King, Milo M. K. Martin, and Jonathan M. Smith. USENIX ;login. December 2010, 31–41.

Lessons Learned During the Development of the CapoOne Deterministic Multiprocessor Replay System. Pablo Montesinos, Matthew Hicks, Wonsun Ahn, Samuel T. King, and Josep Torrellas. Workshop on the Interaction Between Operating Systems and Computer Architecture. June 2009.

Reflex: A Real-World TB* Implementation. Matthew Hicks. University of Illinois at Urbana-Champaign, Masters Thesis. August 2008.

Professional Service

Program Committee:

- 2017 ACM Conference on Computer and Communications Security (CCS)
- 2017 IEEE Symposium on Security and Privacy (Oakland)
- 2017 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)
- 2017 International Workshop on Energy Harvesting & Energy-Neutral Sensing Systems (ENSsys)

Work Experience

Technical Staff **September 2016 to Present**

Lead research on a range of low-level security topics.

Lecturer **September 2015 to August 2016**

- **EECS 388** **W16**
  Introduction to computer security.
- **EECS 183** **F15**
  Introduction to programming (C++ and Python) for non-engineering majors.

Postdoctoral Research Fellow **May 2013 to 2015**

- **Energy-dictated Computing** **September 2013 to Present**
  Batteries prevent the realization of the dream of computational dust. As computation moves to smaller cores, batteries account for a greater portion of the size, cost, and maintenance of embedded systems. Energy harvesting removes the need for a battery, but creates a new problem of an unpredictable power source. My work in this area seeks to build automatic program transformations and new architectures so that programs run correctly and efficiently (i.e., minimize the number of power cycles to complete a task) across power cycles. I am building systems using LLVM, miBench, and a ARM Cortex-M0+ core running on a Xilinx Virtex-5 FPGA.
- **Approximate Computing** **September 2013 to Present**
  Many classes of applications accept a wide range of outputs as correct. This means that it is possible to allow errors in program outputs while still producing outputs that the application deems acceptable. Reducing guard bands to allow for some errors can reduce power consumption or increase performance. In this area, I am building an approximate experimentation platform that consists of both approximate DRAM and an approximate processor. For these projects, I am using a Xilinx Virtex-5 FPGA, a OpenRISC OR1200 processor, and an MSP430 development board.
- **Malicious hardware** **May 2007 to Present**
  This research focuses on both the attack and defense directions of malicious hardware. The attack portion attempts to find and implement interesting and novel attacks in a SoC environment. The defense portion of this research focuses on developing practical and automatic techniques and tools for detecting and removing malicious circuits from HDL code.

Graduate Researcher **May 2006 to 2013**

- **Imperfect hardware** **January 2011 to Present**
  Hardware is not perfect, whether it be design-time bugs created by HDL authors or run-time faults created by an attacker. My research in this area strives to protect unsuspecting software from the implications of imperfect hardware.

- **Malicious hardware** **May 2007 to Present**
  This research focuses on both the attack and defense directions of malicious hardware. The attack portion attempts to find and implement interesting and novel attacks in a SoC environment. The defense portion of this research focuses on developing practical and automatic techniques and tools for detecting and removing malicious circuits from HDL code.

- **Deterministic replay on multiprocessors** **Summer 2008**
  The goal of this research project is to combine hardware and software together in a deterministic replay system for multiprocessors that has the performance advantages of hardware-only replay, but also has the small log size and ability to record a subset of processes provided by software-only replay.

- **Hardware support for real-time systems** **Aug 2007 to May 2008**
  This research focuses on the FPGA implementation of a real-time scheduling algorithm that, while theoretically optimal, is impractical on today's processors. We construct A SoC using the FPGA's built-in processor, running a lightly modified RTOS with the scheduling and other RTOS services offloaded to hardware.

- **FPGA+DSP communications supercomputer** **May 2006 to Jan 2008**
  The goal of this work was to build, from the ground up, a supercomputer for multi-channel environments like packet processing or call processing. The supercomputer relies on FPGAs for packet routing and DSPs for calculation and up and down butterfly networks for deadlock free routing. The supercomputer is node-based and channels are added by attaching more nodes. I was tasked with PCB design and all FPGA development, except the crossbar switch.