Investigating Energy and Security Trade-offs in the Classroom With the Atom LEAP Testbed

Peter A. H. Peterson Digvijay Singh William J. Kaiser
Peter L. Reiher
{pahp, reiher}@cs.ucla.edu digvijay@ucla.edu kaiser@ee.ucla.edu

4th Workshop on Cyber Security Experimentation and Test

August 8, 2011

UCLA

Overview

We present a new measurement tool and describe its use for an undergraduate research seminar investigating energy and security tradeoffs.

- Introducing the Atom LEAP
- Undergraduate Research: Security vs. Energy
- Lessons Learned

Introducing Atom LEAP

Introducing Atom LEAP

The Atom LEAP is an energy measurement platform which is...

- Component Level (HDD, RAM, CPU, USB, PSU)
- High granularity (10,000 floating point samples per second)
- Event Synchronized with Energy Calipers

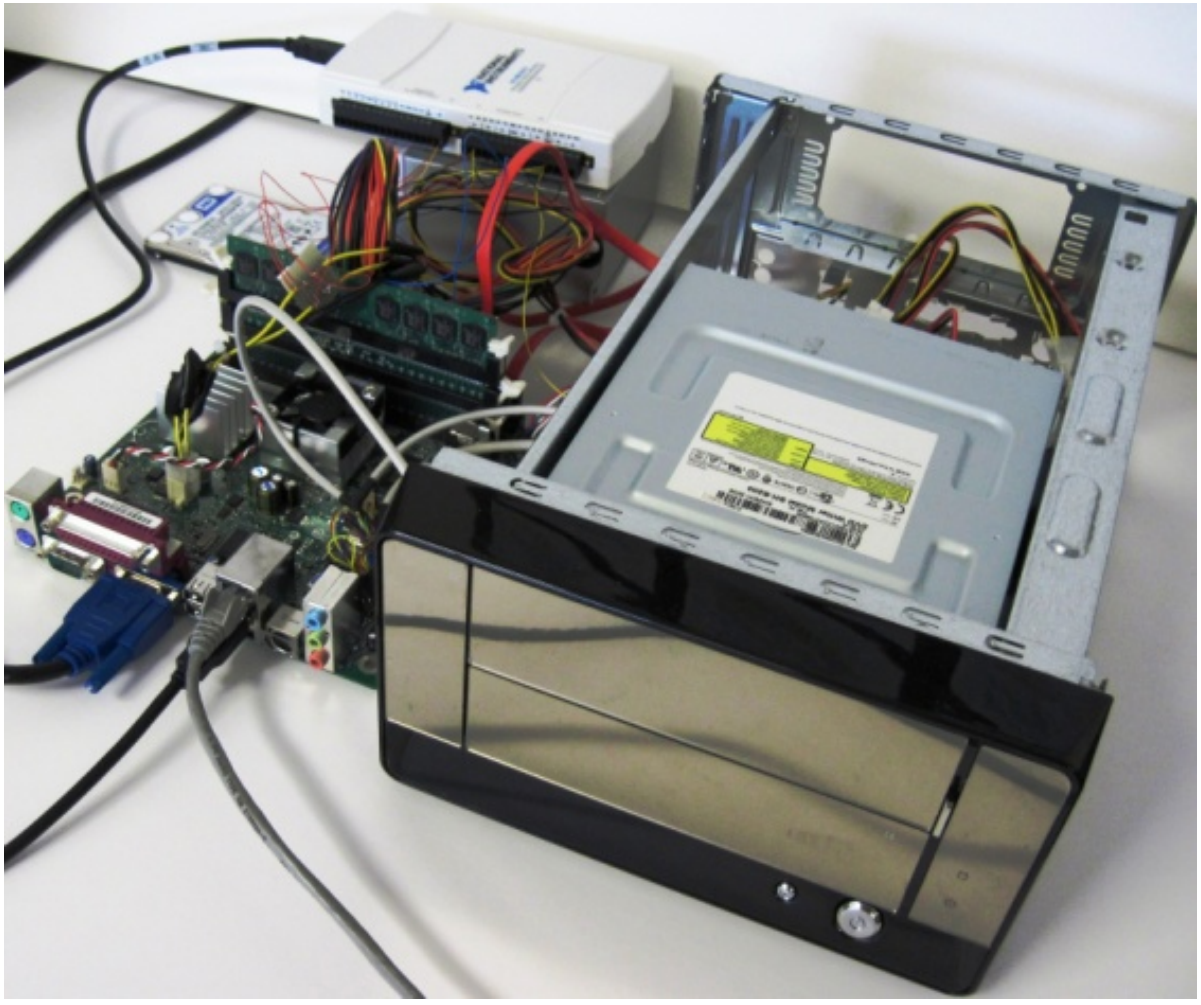
The LEAP is flexible and portable:

- "Off the shelf" hardware
- Open-source/Free Software stack
- "Easy" to construct (Even I can do it!)
- "Inexpensive" — about $1,500 (mostly the DAQ)
- Self-contained

Atom LEAP Components

- Intel Atom N330 and Motherboard
- Risers and sense resistors
- National Instruments USB DAQ
- Synchronization instrumentation
- Software stack

Atom LEAP: Exploded View



UCLA

Synchronized Energy Calipers

Energy calipers allow energy measurement of very small time spans.

- A sync signal is linked to the CPU's Time Stamp Counter (TSC)
- This signal is sampled while workload is measured
- Workload instrumentation records the TSC before and after an event with inline assembly or a command line utility
- Post-hoc analysis identifies energy samples taken during named events
- Data can be averaged or viewed as a time series

Synchronized Energy Calipers

Calipers are available for user and kernel code:

- Kernel prints TSC values to the syslog
- User code prints TSC values to a log file
- All logging is done to a RAM disk to reduce overhead (although this obviously increases RAM usage)

Software Stack

Familiar and powerful environment:

- Runs on Linux
- Can test virtually any software with minimum instrumentation
  - Inline assembly compiled into source code
  - Or, `getticks` utility executed between binary workloads
- Testing scripts facilitate consistent repetitions and batched jobs with reboots
- Report format named by workload, separated into job components, and easily imported into analysis tools

Portable and Flexible Framework

The LEAP model is applicable to a wide variety of hardware. It requires only:

- Stable, accessible clock source like a TSC
- Instrumentation harness (wires, resistors, and DAQ)
- Software tools – automating workloads, scripting, etc.

We have already constructed:

- Atom LEAP
- Mobile LEAP
- DSP LEAP
- Dual Xeon LEAP

Active Testing and Development

Ongoing research:

- analysis of self-measurement cost
- analysis of sample rate effects
- improved instrumentation code
- calibration workloads

Stay tuned for information about a pilot program.

Undergraduate Research with Atom LEAP

UCLA

Undergraduate Research with Atom LEAP

Case study:

- UCLA CS 188: Undergraduate Research Seminar
- 25 graduates and undergraduates
- 10-week quarter
- Security-oriented projects
- Atom LEAP

Our goals:

- Using LEAP to investigate Security vs. Energy Trade-offs
- Teach performance measurement and analysis
- Interactive group meetings instead of lectures
- Emphasis on overall process and experience rather than novel results

UCLA

Introducing the Atom-Powered zPad by Banana Computer

As employees of Banana Computer's Development Lab, students were assigned to workgroups performing introductory investigations into five energy and security related concept areas for the new device:

- Disk Encryption
- Network Encryption
- Power/Security Posture
- Virtualization/Isolation
- Offloading Computation

I'll briefly discuss each area, along with some limited quantitative results.

But First, A Disclaimer

The following nuggets are meant more as teasers than solid gold "results." Several issues combined to affect outcomes:

- Experiment design and workload choice
- Time constraints of quarter
- Varying expertise of students
- Experimental practice

We'll discuss some of these issues in the Lessons Learned portion of the talk.

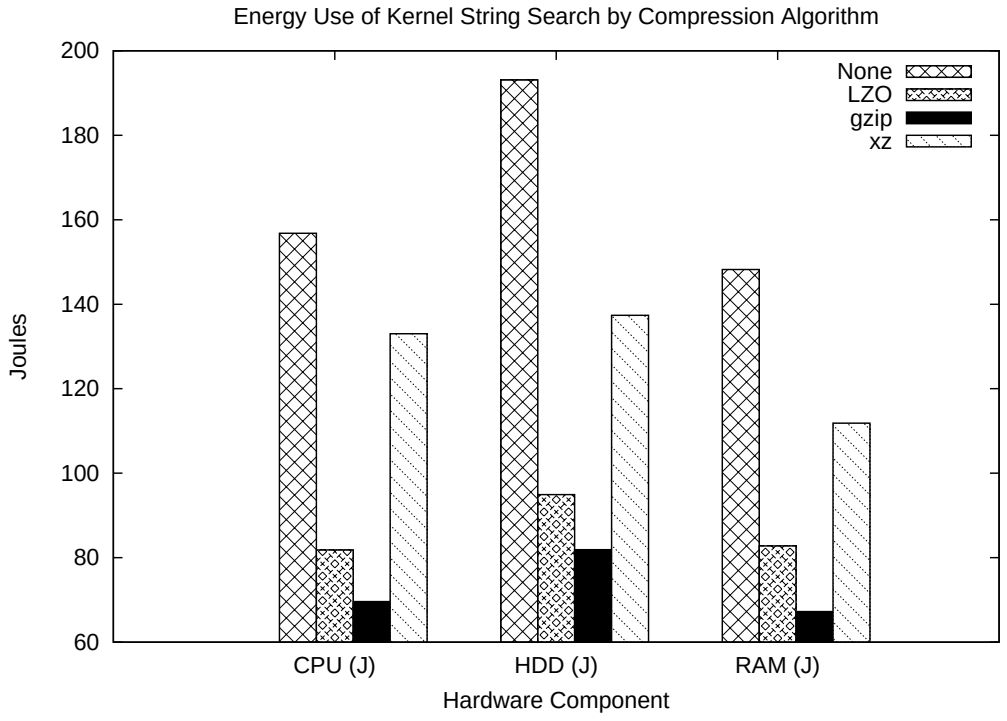
1. Disk Encryption

- How much energy does Full Disk Encryption cost?
- Should it be done in hardware or software?

FDE disk did not arrive in time, so students investigated whether compression could reduce the cost of disk encryption.

- Compared various filesystems and compression algorithms
- Against various workloads

1. Disk Encryption: Food for thought



Does compression save energy? If so, how much?

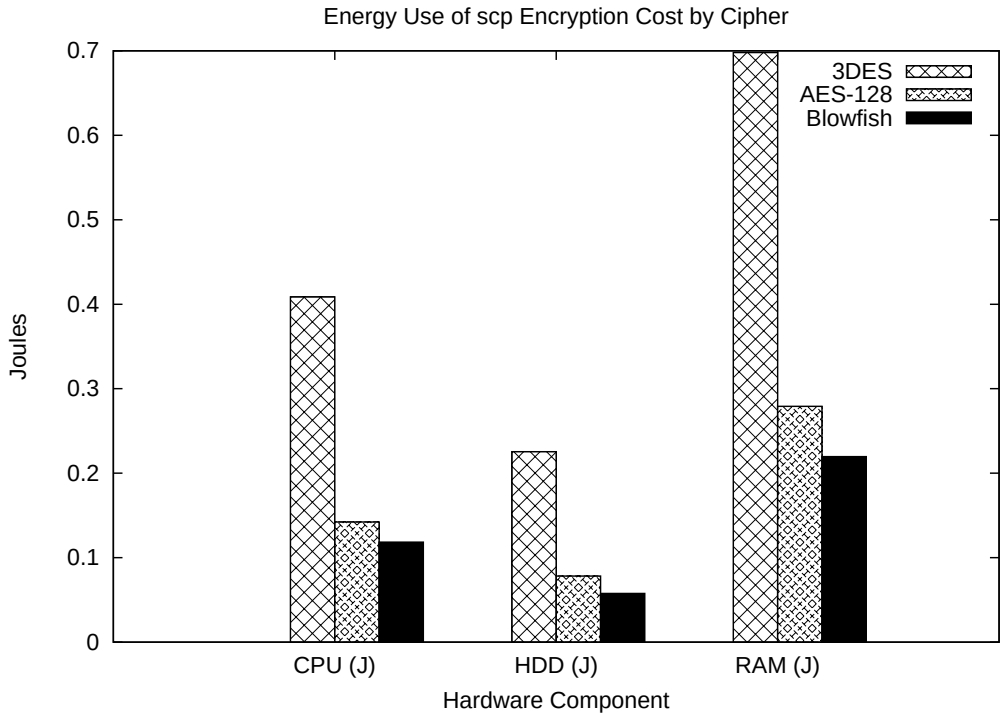
- Compression significantly reduces cost of encrypted filesystem
- Middle ground algorithm (gzip) is best

UCLA

2. Network Encryption

- How much does encryption cost change by algorithm?
- What if we chose algorithm based on current threat?
- Students chose to engineer cipher changing in SSH
- Workloads included basic microbenchmarks as well as expect-scripted interactive sessions and file transfers.

2. Network Encryption: Food for thought



Does SCP cipher choice have an energy cost?

- Both AES and Blowfish are significantly cheaper than 3DES
- Blowfish is the winner (but this was only significant for RAM)

UCLA

3. Sandboxing

- How much energy do various sandboxing techniques cost?
  - VirtualBox
  - User-Mode Linux
  - chroot jails
- What isolation is provided by each technique?
- Students ran file utilities and GUI workloads scripted with the Linux Desktop Testing Project (LDTP)

3. Sandboxing: Food for thought

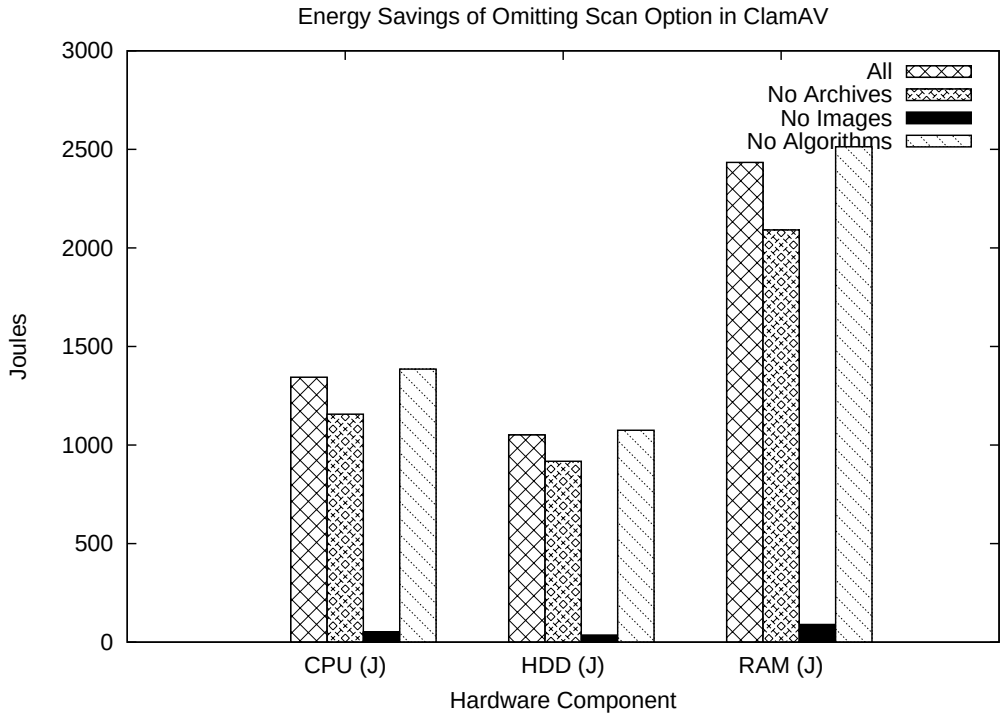
The Atom N330 Does not have hardware virtualization support!

- 60-600% *penalty* to use VirtualBox! (Even with kernel plugin and multiple cores)
- chroot jails inexpensive, but of limited value
- Students could not get User-Mode Linux to work

4. Dynamic Power/Security Posture

- What if OS added energy dimension to Security and Data Sensitivity "zones"?
- In high-security areas, the zPad could cut back on paranoid security
- Or reschedule latency-tolerant tasks for later
- Would this save energy?
- What would such a facility look like?

4. Power/Security Posture: Food for thought



Energy-aware virus scanning

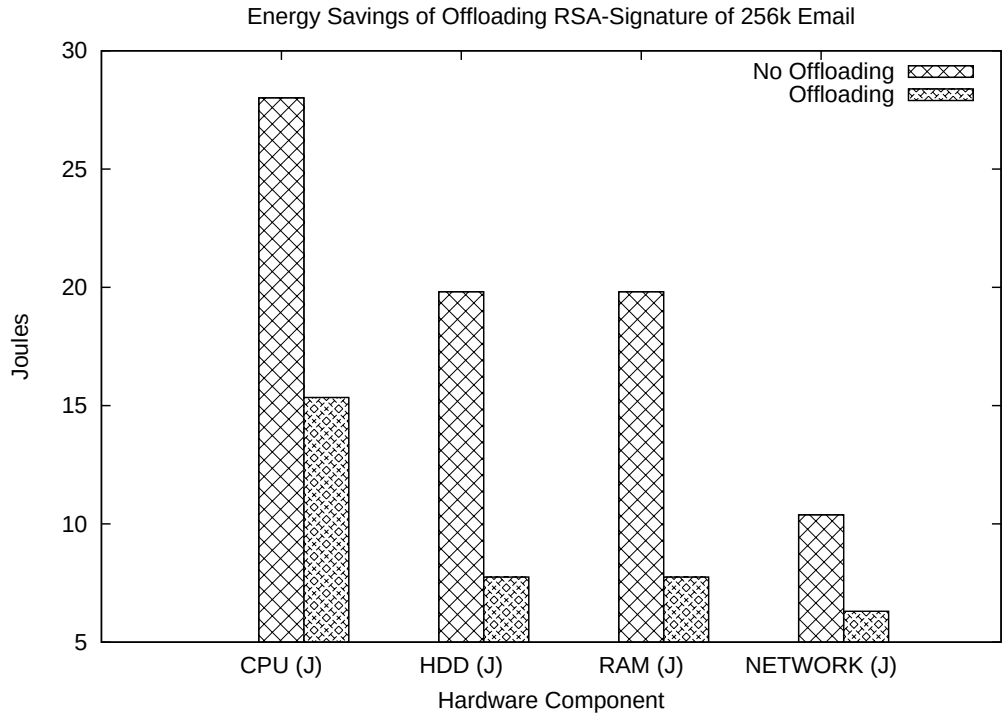
- Students created the PowerSecZone daemon and API
- Modified ClamAV to behave differently based on status
- Number of files scanned more significant (in their test) than content or algorithm

UCLA

5. Offloading Security

- Can the zPad extend battery life by offloading security tasks?
- Or does the offloading cost more than it saves?
- Tested email signing, bittorrent, and more

5. Offloading: Food for thought



Offloading cryptographic signing to a compute server via WiFi

- Saved a tremendous amount of power — around 40-50%
- Even while paying to encrypt wifi with SSH
- Not just shorter runtime – disk, RAM, and net (USB) used less watts

UCLA

Lessons Learned

The course and Atom LEAP are a success, but there is much work to do!

- Course design and management
- Research value vs. practical issues
- Statistics and experimental practice
- LEAP Development

Lessons: Course Design

- Atom LEAP technology worked really well
- Designing projects and assigning students was a good choice
- Group meetings a lot like actual grad school experience
- Five *different* groups require a lot of oversight

Lessons: Research Value vs. Practical Issues

- Help students prepare to succeed despite major setbacks
- Decide if you want rock solid results or broader educational experience
- Quarters are short and students (allegedly) have other classes

Lessons: Statistics and Experimental Practice

- The class definitely taught why evaluation is hard
- We worked hard to try and control parameters and use real software
- Not all students learned good experimental practice and this wasn't obvious until it was too late
- Be explicit about experimental requirements:
  - *require* students to script workloads for repeatability
  - *require* analysis of workload scripts with students (not just experimental plan)
  - *require* statistical analysis, don't just recommend it
- Test early, test often!

Atom LEAP Pilot Program

We are improving LEAP technology and are preparing a distribution of the platform so that other researchers can build their own. We're creating documentation, packaging our tools, and preparing some community resources. To help iron out some kinks, we are considering a small pilot program. Interested? Email: `pahp@cs.ucla.edu`

UCLA

Acknowledgments

- Intel, NSF
- Sean Peisert (shepherd)
- CSET committee, reviewers, USENIX
- Dr. William J. Kaiser and Digvijay Singh (primary LEAP developers)

UCLA