DOI:10.1145/1785414.1785417

Don't Ignore Security Offshore, or in the Cloud

Moshe Y. Vardi's Editor's Letter "*Globalization and Offshoring of Software* Revisited" and Dave Durkee's "Why Cloud Computing Will Never Be Free" (both May 2010) failed to address security risks. Vardi's headline promised an update on the questions raised by increased globalization of outsourced software development. Though I knew his main focus was on the economic impact of global outsourcing, I was still disappointed there was no mention of the security challenges posed by the global supply chain for software. Such challenges have prompted the U.S. Departments of Defense and Homeland Security, the SAFECode consortium, and numerous other organizations to commit significant effort to combating threats posed by software of unknown pedigree and provenance, including individual and state-sponsored "insider threats" (such as implanted malicious logic, backdoors, and exploitable vulnerabilities), particularly when developed offshore. See the Government Accountability Office's *Defense Acquisitions: Knowledge of Software Suppliers Needed to Manage Risks* (http://www.gao.gov/new.items/d04678.pdf) and the *Report of the Defense Science Board Task Force on Mission Impact of Foreign Influence on DOD Software* (http://www.acq.osd.mil/dsb/reports/ADA486949.pdf). Though both focus on software used by DoD, the security issues apply to any organization that relies on outsourced software for critical business or mission functions.

Meanwhile, in an otherwise admirable assessment of the strengths and weaknesses of the cloud computing model of outsourced IT-as-a-service, Durkee likewise failed to mention potential consequences of cloud providers not protecting outsourced computing infrastructure against hackers and malicious code. For example, when discussing transparency, he overlooked the fact that no cloud provider allows its customers to implement intrusion detection or security monitoring extending into the management-services layer behind virtualized cloud instances. Moreover, these customers have learned not to expect their providers to deliver detailed security-incident, vulnerability, or malware reports.

The management-service layer provides a back channel through which the content of each cloud instance is accessible, not only by providers, but by any attacker able to hack into or implant a kernel-level rootkit. Once "in," the attacker is positioned to exploit the back channel to manipulate or even make full copies of all cloud instances hosted on the compromised platform. Even if customers manage to get their providers to agree to service-level agreements (SLAs) sti pulating a high level of vigilance, reporting, and protection below the cloud-instance layer, the management-services layer remains an inherent weakness that should concern anyone looking to host "in the cloud" the kinds of critical applications Durkee explored.

Karen Mercedes Goertzel,
Falls Church, VA

Author's Response:

I strongly agree with Goertzel's sentiment and appreciate her raising this very important issue. The executive summary of the 2006 Globalization and Offshoring Report *said: "Offshoring magnifies existing risks and creates new and often poorly understood or addressed threats to national security, business property and processes, and individuals' privacy. While it is unlikely these risks will deter the growth of offshoring, businesses and nations should employ strategies to mitigate them." The* Report's *Chapter 6, "Offshoring: Risks And Exposures," covered the risks at length.*

Moshe Y. Vardi, Editor-in-Chief

Author's Response:

As with performance and uptime, cloud security is determined by the necessity of meeting the terms of SLAs as demanded by customers. As they mature, they will demand even more from their providers' SLAs by agreeing to industry-standard audits and certifications that ensure they get the security they need, a topic that is a great starting point for another article.

Dave Durkee, Mountain View, CA

Up in the Air

Describing the network effects of a cloud strategy, particularly when it involves SaaS platform efficiency, in his "Technology Strategy and Management" Viewpoint "Cloud Computing and SaaS as New Computing Platforms" (Apr. 2010), Michael Cusumano said that major cloud hosts, including Amazon, Google, and Salesforce, generally rely on detailed SLAs to guarantee security and other parameters for their hosted customers. However, many such hosts, including Amazon SimpleDB and Google Apps, agree to SLAs involving only, perhaps, performance degradation limits and availability of a given service. If cloud-related SLAs fail to include more specific parameters, the cloud infrastructure risks closing itself to new, innovative services due to its lack of dependable guarantees.

Burkhard Stiller and **Guilherme Machado**, Zürich, Switzerland

Diversity Factor

Richard Tapia's inspiring Viewpoint "Hiring and Developing Minority Faculty at Research Universities" (Mar. 2010) said that looking for the next Gauss or Turing is not necessarily the key criterion in all CS faculty searches. I have sometimes sensed confusion between the notion that research excellence drives academic success (it does and should) and what might be called the "additive argument," or belief that maximizing the potential research stature of every new hire automatically maximizes a department's overall excellence in research. I read Tapia's section on reexamining search criteria to mean this is not always the case. I concur, convinced that the effects of talent are not simply additive.

It ought to go without saying that the goal of diversity of gender or ethnic origin does not generally conflict with excellence in research. For instance, in recent years my department has interviewed several women candidates who were uniformly superior to their male counterparts.

However, in specific faculty searches it may be that the potential research stature of a certain white male candidate is perceived as exceeding that of a certain female or minority candidate. The latter may be stellar, but the former's intellectual light shines just a bit brighter. If the discrepancy is comparable to the rather high level of uncertainty inherent in measuring a candidate's potential, some may invoke the additive argument.

However, this argument seems to rest on two questionable assumptions: departmental excellence (however measured) is the arithmetic sum of the individual levels of excellence of its faculty members; and the success of an individual researcher is independent of the surrounding environment.

Both are wrong. Excellence in research (individually or across a department) is a nonlinear function of interdependent factors. For instance, in a department that makes itself attractive to a broader pool of graduate students through the composition of its faculty, all researchers benefit from the resulting potentially improved quality of the department's student body. This also holds when attracting new colleagues, including so-called superstars. When female or minority candidates are at, say, the top of the list in a particular search, they (like everybody else) also consider a department's environment when choosing which job offer to accept. Moreover, a more welcoming, collegial, diverse faculty often leads to better and more frequent collaboration, as well as to more vibrant research.

The question is not whether to compromise between excellence and diversity but how best to foster excellence, with diversity a part of the equation.

Carlo Tomasi, Durham, NC

Wrong Side of the Road

In his Editor's Letter "Revisiting the Publication Culture in Computing Research" (Mar. 2010), Moshe Y. Vardi said computer science is "the only scientific community that considers conference publications as the primary means of publishing our research results," asking, "Why are we the only discipline driving on the conference side of the 'publication road?'"

As an old timer, I can say that in the early days, there was a belief (conceit might be a better word) that the field's pace of discovery was happening so quickly that only conferences, with subsequent prompt publication of proceedings, could communicate results in a timely manner. As a corollary, the traditional peer-reviewed published literature review fell behind, as it was relieved of temporal pressure through the published proceedings.

These days, the pace of discovery in the biological sciences, including molecular biology, genomics, and proteomics, far exceeds that of computer science. Yet the gold standard of publication in archival journals continues. It is the ultimate irony that computer science, along with various disciplines in the physical sciences, employs the tools developed by computer scientists to ensure timely dissemination of research results through the online editions of their publications. *Science*, *Nature*, *Cell*, and other leading journals routinely present their most important articles in online form first. If, perhaps, computer science would make greater use of its own tools, the shoemaker's children would no longer go barefoot, and published proceedings would fade into its proper historical niche.

Stuart Zimmerman, Houston, TX

More to Celebrate in RDBMS History

Gary Anthes offered good reporting but also some serious errors concerning pre-RDBMS history in his news article "Happy Birthday, RDBMS!" (May 2010), saying "In 1969, an ad hoc consortium called CODASYL proposed a hierarchical database model built on the concepts behind IMS. CODASYL claimed that its approach was more flexible than IMS, but it still required programmers to keep track of far more details than the relational model did."

Please compare with the following basic facts as reported in Wikipedia: "In 1965 CODASYL formed a List Processing Task Force. This group was chartered to develop COBOL language extensions for processing collections of records; the name arose because Charles Bachman's IDS system (which was the main technical input to the project) managed relationships between records using chains of pointers. In 1967 the group renamed itself the Data Base Task Group and in October 1969 published its first language specifications for the network database model, which became generally known as the CODASYL Data Model."

The Integrated Data Store (IDS) has been in continuous productive use since 1964, running first on GE 200 computers. In 1966, it began supporting a nationwide, 24/7, order-entry system (OLTP). And in 1969, running on the GE 600, it began supporting a shared-access (OLTP) database, complete with locks, deadlock detection, and automatic recovery and restart.

IBM did not release its IMS/360 (Information Management System) based on the hierarchical data model until September 1969 when future relational databases were still just a gleam in Ted Codd's eye.

B.F. Goodrich received the IDS source code from GE in 1964, renaming it the Integrated Database Management System, or IDMS, when rewritten for the IBM 360 (1969–1971). IDMS was acquired (1973) and marketed worldwide by Cullinane (later Cullinet). IDMS was acquired (1989) by CA (formerly Computer Associates), which still actively supports it worldwide on more than 4,000 IBM mainframes. British Telecom and the Brazilian government are the best-known IDMS users, rated, in 2005, the second- and third-largest OLTP systems in the world.

For more, please see the refereed papers on IDS, IMS, IDMS, and other DBMS products in *IEEE Annals of the History of Computing* (Oct.–Dec. 2009) special issue on "Mainframe Software: Database Management Systems." A future issue is planned to cover more recent RDBMS history.

Charles W. (Charlie) Bachman, Lexington, MA, ACM Turing Award 1973

Communications welcomes your opinion. To submit a Letter to the Editor, please limit your comments to 500 words or less and send to letters@cacm.acm.org.