

NICKOLAI ZELDOVICH

Stanford University, Computer Science Department
353 Serra Mall, Room 288
Stanford, CA 94305-9025

Phone: (650) 996-4201
nickolai@cs.stanford.edu
<http://www.scs.stanford.edu/~nickolai>

EDUCATION

- 2002–2007 **Stanford University** Stanford, CA
01/2008 Ph.D. in Computer Science
Thesis title: *Securing Untrustworthy Software Using Information Flow Control*.
Advisor: David Mazières.
- 1998–2002 **Massachusetts Institute of Technology** Cambridge, MA
06/2002 M.Eng. in Electrical Engineering and Computer Science
Thesis title: *Concurrency Control for Multi-Processor Event-Driven Systems*.
Advisor: Robert Morris.
- 06/2002 S.B. in Electrical Engineering and Computer Science, minor in Mathematics

RESEARCH INTERESTS

Operating systems, distributed systems, security, and networking.

RESEARCH PROJECTS

- 2005–present **HiStar: a secure operating system [2]**. Led the HiStar project on designing and developing a new operating system that allows applications to minimize the amount of trusted code. HiStar allows applications to specify precise data security policies by specifying how different information can flow through the system. As a result, small amounts of trusted code can reason about the security of large, complex, and potentially buggy applications. HiStar enforces precise information flow restrictions by breaking down and refactoring traditional OS abstractions into six basic types of objects and a small number of operations that make all information flow explicit.
- HiStar solves a number of challenges. In particular, HiStar tracks information flow dynamically without allowing the tracking mechanism itself to leak information. By decoupling resource allocation and revocation from all other forms of resource access, HiStar also allows a system administrator to manage resources without any inherent superuser privileges to read and write all data in the system.
- HiStar's abstractions are flexible enough to implement a self-hosting Unix-like environment in an untrusted user-level library. At the same time, HiStar supports novel, highly-secure applications side-by-side with traditional Unix code, such as an entirely untrusted login process, or privacy-preserving untrusted virus scanners.
- 2006–present **Security in distributed systems [1]**. Designed and implemented DStar, a decentralized network protocol for enforcing information flow control in a distributed system. By controlling information flow, DStar allows building secure distributed applications from largely untrusted code, such as a scalable web server that has little trusted code and no fully-trusted machines or components.
- DStar's decentralized environment poses a number of unique challenges, including trust and resource allocation, whose solution is taken for granted on a single machine. The

key idea in DStar is to use self-certifying names to specify information flow restrictions. This allows individual machines to determine who is trusted to handle what data without any external communication, which could otherwise leak information.

2007–present **Hardware support for security [8].** Co-designed Loki, a hardware architecture for minimizing trusted code, and re-architected HiStar to take advantage of Loki. Loki directly enforces security policies on words of physical memory in hardware, instead of relying on indirect mechanisms, such as page tables, to enforce security. A novel three-layer kernel architecture allows HiStar to reduce the amount of fully-trusted code by a factor of two on Loki, starting from an already-small kernel of under 20,000 lines of code. A small *security monitor*, running with full privileges, enforces traditional read/write memory access control. The rest of HiStar’s kernel code executes on top of the security monitor and enforces information flow control but cannot violate the monitor’s simpler security guarantees. A full-system FPGA prototype of Loki achieves good performance running a variety of workloads in HiStar’s Unix-like environment.

2006–present **Security verification.** Ongoing work in collaboration with a number of research groups at Stanford to verify the security of HiStar, DStar, and Loki. HiStar’s clear definition of security in terms of information flow control allows precisely specifying what it means to be secure, and what it means for security to be broken. This formal definition enables a number of approaches to verifying security, with promising results. Using model checking, we have proven that a subset of HiStar’s kernel interface is secure. We used static analysis to ensure that certain safety properties hold in the implementation of the HiStar kernel. Finally, we are examining the behavior of HiStar’s kernel code on all possible inputs using symbolic execution. This allows verifying a wide range of properties; currently the focus is on test coverage, while the long-term goal is proving the security properties of the information flow control mechanism.

2002–2005 **Virtual appliances [3, 4, 5].** Designed and implemented systems that used virtual machines to solve problems of user mobility, software distribution, and system management [3]. Treating the entire virtual machine as the unit of mobility, distribution, and management solves a number of problems, such as missing or mismatched dependencies, broken software installs, and unreliable updates. Virtual machine monitors also provide a reliable recovery mechanism even when the virtual machine has been compromised.

Co-founded a company called moka5 to commercialize the research project’s ideas and prototypes in the context of desktop software management.

Worked on caching policies and prefetching algorithms for downloading virtual machine disks over the network. Designed and implemented a system for measuring interactive performance of desktop environments [4] to evaluate the resulting performance. Developed an object-oriented language to describe, configure, and distribute virtual machines and networks of virtual machines [5].

2001–2002 **Concurrency control [6, 7].** Designed and implemented a simple mechanism for exposing computational concurrency in event-driven programs, by associating a *color* with every event callback to represent the accessed data. Existing programs can incrementally color their callbacks to take advantage of multi-processor systems. Achieved significant performance improvements for an event-driven file server on a multi-processor machine, by modifying 90 lines of code to parallelize cryptographic operations.

WORK EXPERIENCE

- 10/2007–present **Postdoctoral Scholar.** CS Department, Stanford University Stanford, CA
- 05/2005–09/2005 **Co-founder.** SkyBlue Technologies (now moka5) Redwood City, CA
- 09/2002–09/2007 **Research Assistant.** CS Department, Stanford University Stanford, CA

- 06/2001–08/2002 **Research Assistant.** PDOS Research Group, MIT LCS Cambridge, MA
- 06/2000–08/2000 **Embedded Software Developer.** Kaveri Networks Sunnyvale, CA
Designed and developed network protocols for low-power Internet-connected devices.
- 11/1999–02/2004 **Student Programmer.** Athena Server Operations Group, MIT Cambridge, MA
Helped develop and maintain a variety of computing services at MIT, including the AFS file system, web services, and remote login.
- 06/1999–08/1999 **Research Intern.** Naval Research Laboratory Washington, DC
Incorporating just-in-time signaling protocols in optical cross-connect switches.
- 05/1998–08/1998 **Undergraduate Researcher.**
Computer Vision Lab, University of Central Florida Orlando, FL
Worked on video segmentation, shot segmentation and similarity, and violence detection, using skin detection and optical flow algorithms.
- 08/1997–09/2003 **Systems Administrator.** Craig’s Data Exchange Mount Dora, FL
Helped manage all technical aspects of a medium-size Internet service provider.
- 06/1997–08/1997 **Research Intern.** EE Department, Princeton University Princeton, NJ
Implemented cyclic memory access optimizations for DSP chips in the SUIF compiler.
- 06/1996–08/1996 **Web Application Developer.** University of Central Florida Orlando, FL
Developed dynamic, database-driven web applications.

TEACHING EXPERIENCE

- 12/2007 **Guest Lecturer.** Operating Systems (CS 140), Stanford University.
- 05/2007 **Guest Lecturer.** Advanced Topics in Operating Systems (CS 240), Stanford University.
- 01/2007–03/2007 **Course Assistant.** Distributed Systems (CS 244b), Stanford University. Developed and graded lab assignments that involved students building parts of a network file system replicated using Paxos. Taught lab section. Helped students understand papers covered in course.
- 09/2005–12/2005 **Course Assistant.** Advanced Operating Systems Implementation (CS 240c), Stanford University. Helped students with lab assignments, which involved developing parts of a small operating system. Answered questions about OS research papers covered in class.

PROFESSIONAL ACTIVITIES

- Journal Reviewer:** Journal of Systems and Software (JSS), 2007.
- Conference Reviewer:** VEE 2008, PACT 2007, SOSP 2007, IEEE Security and Privacy 2007, OSDI 2006, SOSP 2003.

REFEREED CONFERENCE PUBLICATIONS

- [1] Nickolai Zeldovich, Silas Boyd-Wickizer, and David Mazières. Securing distributed systems with information flow control. In *Proceedings of the 5th Symposium on Networked Systems Design and Implementation (NSDI)*, San Francisco, CA, April 2008. To appear.
- [2] Nickolai Zeldovich, Silas Boyd-Wickizer, Eddie Kohler, and David Mazières. Making information flow explicit in HiStar. In *Proceedings of the 7th Symposium on Operating Systems Design and Implementation (OSDI)*, pages 263–278, Seattle, WA, November 2006.

- [3] Ramesh Chandra, Nickolai Zeldovich, Constantine Sapuntzakis, and Monica Lam. The Collective: A cache-based system management architecture. In *Proceedings of the 2nd Symposium on Networked Systems Design and Implementation (NSDI)*, pages 259–272, Boston, MA, May 2005.
- [4] Nickolai Zeldovich and Ramesh Chandra. Interactive performance measurement with VNCplay. In *Proceedings of the USENIX 2005 Annual Technical Conference, FREENIX track*, pages 189–198, Anaheim, CA, April 2005.
- [5] Constantine Sapuntzakis, David Brumley, Ramesh Chandra, Nickolai Zeldovich, Jim Chow, Jim Norris, Monica S. Lam, and Mendel Rosenblum. Virtual appliances for deploying and maintaining software. In *Proceedings of the 17th USENIX Large Installation System Administration Conference*, pages 181–194, San Diego, CA, October 2003.
- [6] Nickolai Zeldovich, Alexander Yip, Frank Dabek, Robert T. Morris, David Mazières, and M. Frans Kaashoek. Multiprocessor support for event-driven programs. In *Proceedings of the USENIX 2003 Annual Technical Conference*, pages 239–252, San Antonio, TX, June 2003.

REFEREED WORKSHOP PUBLICATIONS

- [7] Frank Dabek, Nickolai Zeldovich, M. Frans Kaashoek, David Mazières, and Robert Morris. Event-driven programming for robust software. In *Proceedings of the 10th ACM SIGOPS European Workshop*, pages 186–189, September 2002.

IN SUBMISSION

- [8] Hari Kannan, Nickolai Zeldovich, Michael Dalton, and Christos Kozyrakis. Architectural support for minimizing trusted code. In submission, November 2007.

INVITED TALKS AND PRESENTATIONS

- 12/2007 **Stanford Information Networks Group Seminar** Stanford, CA
Securing Untrustworthy Software Using Information Flow Control
- 10/2007 **Stevens Institute of Technology** Hoboken, NJ
Securing Untrustworthy Software Using Information Flow Control
- 08/2007 **Sun Labs** Menlo Park, CA
Securing Untrustworthy Software Using Information Flow Control
- 05/2007 **University of California, Berkeley Systems Lunch Seminar** Berkeley, CA
Securing Untrustworthy Software Using Information Flow Control
- 04/2007 **Stanford Clean Slate Network Research Seminar** Stanford, CA
Distributed Information Flow Control
- 03/2007 **TRUST Site Visit Poster Session** Berkeley, CA
Making Information Flow Explicit in HiStar
- 11/2006 **2006 Usenix OSDI Conference** Seattle, WA
Explicit Information Flow in the HiStar OS
- 10/2006 **Kryptos Study Tour** Stanford, CA
Making Information Flow Explicit in HiStar
- 04/2005 **2005 Usenix Annual Technical Conference** Anaheim, CA
Interactive Performance Measurement with VNCplay
- 06/2003 **2003 Usenix Annual Technical Conference** San Antonio, TX
Multiprocessor Support for Event-Driven Programs

05/2003 **Stanford Computer Forum Poster Session**Stanford, CA
Virtual Appliances

AWARDS

06/2001 George C. Newton Annual Prize for best undergraduate laboratory project at MIT. The project was a networked thermometer, which involved connecting an 8-bit microcontroller to an Ethernet adapter and temperature sensor, and developing a network stack and web server in assembly.

PATENTS

Monica Lam, Constantine Sapuntzakis, Ramesh Chandra, Nickolai Zeldovich, Mendel Rosenblum, Jim Chow, and David Brumley. "Virtual Appliance Management." U.S. Patent Application No. 11/007,911. Pending, filed December 2004.

PROFESSIONAL AFFILIATION

Member of ACM (1998), Usenix (2002).

PERSONAL INFORMATION

United States citizen. Fluent in Russian. Date of birth 11/11/1981. Married.

REFERENCES

Prof. David Mazières
Stanford University CS Department
353 Serra Mall, Room 290
Stanford, CA 94305-9025
(650) 723-8777
kolyarec@nospam.scs.stanford.edu

Prof. Monica Lam
Stanford University CS Department
353 Serra Mall, Room 307
Stanford, CA 94305-9030
(650) 725-3714
lam@stanford.edu

Prof. Dawson Engler
Stanford University CS Department
353 Serra Mall, Room 314
Stanford, CA 94305-9030
(650) 723-0762
engler@stanford.edu

Prof. Robert Morris
MIT Computer Science and AI Lab
32 Vassar St., Room 32-G972
Cambridge, MA 02139
(617) 253-5983
rtm@csail.mit.edu

Stanford, CA, January 8, 2008