

## Objectives

To keep the bits flowing, in fewer cycles, whilst burning fewer dinosaurs.

I seek expert-level consulting work in systems programming, high-performance computing, or anything else that lets me avoid Java.

## Open Source

See <https://www.nick-black.com/dankwiki/index.php/Hackery> for more details.

I passionately believe in the many societal benefits of open source, and actively participate in the Free Software community.

- Author and maintainer of numerous open source packages. Examples include:
  - *omphalos*, an application for securing and attacking local networks,
  - *growlight*, an application for managing solid-state storage and disk arrays,
  - *libtorque*, a multithreaded, architecture-adaptive event library, and
  - *cubar* and *libcudest*, reverse-engineering tools for NVIDIA’s CUDA.
- Maintained IBM’s NGPT (Next Generation POSIX Threading) kernel patchset.
- Co-maintained the x86info processor identification tool.
- Contributed accepted patches to the Linux kernel (IPv4 stack, hugetlbfs, device drivers), ZoL (ZFS on Linux), ipset, GNU libc, Wireshark (IPv4 analysis),  $\mu$ Clibc, DynamoRIO, OProfile, Vim, Ncurses, GMPC, OpenWRT, iptables, iperf, nmap, dpkg, strace, OpenSSL, APT-secure, Infinality, iproute2, bridge-utils, cpuid, Snort, and other projects.
- “Dynamic iSCSI at Scale: Remote Paging at Google”, invited talk, 2015 Linux Plumbers Conference, Seattle WA.

## Employment and Education

- **2012/01–2013/01, 2018/03–Present: Consulting Scientist, Dirty South Supercomputing (Atlanta, GA)**

Founder

Bespoke consulting on low-level, embedded, and high-performance work including:

- An anti-drone platform including sensor fusion and offensive RF.
- A filesystem targeting NAND, plus a QSPI driver, running in 2KB on bare metal.
- A distributed ledger, including P2P networking and extensive crypto, for health records.
- Work on DynamoRIO to efficiently perform dynamic binary analysis in the presence of JIT.
- LoRa radio, CAN bus reverse engineering, and embedded work for autonomous lawnmowing.

- **2016/10–2018/03: Chief Engineer, Luma (Atlanta, GA)**

Acquired by Newell Brands 2018-01-02

- Designed and implemented C++ system agent for cloud-managed mesh WiFi routers, replacing the existing agent fleetwide.
- Researched and implemented novel solutions for adaptive, cloud-managed meshing on Qualcomm and Linux NL80211 platforms in tandem with cloud-based ML-assisted WiFi optimization.

- **2014/01–2016/10: Senior Software Engineer, Google (New York, NY)**

2x Technical Infrastructure Award

- Systems programming work for the Technical Infrastructure team.
- Work on heterogeneous scheduling and packaging.
- HPC work using C++, CUDA, and assembly language for machine learning.

- **2011/01–2012/02: Senior Compiler Engineer, NVIDIA (Austin, TX)**

- Development on NVIDIA’s OCG (Optimizing Code Generator) for Fermi and Kepler GPUs and Denver CPUs.
- Led design and implementation work on a PTX/SASS/SL/Cg-unifying assembler.

- **2008/08–2010/05: Georgia Institute of Technology, Master of Science in Computer Science**

- Research project: “Epicycles, Flywheels, and (Widening) Gyres: UNIX I/O Slouches Toward Multicore NUMA” (2009). Results include the open *libtorque* library and a 2010 paper, “Portable Multithreaded Continuations for Scalable Event-Driven Programs”.
- Teaching Assistant, “High Performance Computer Architecture” (CS6290—Superscalar, OOO, and manycore microarchitecture).

- **2005/12–2010/05: Principal Engineer, McAfee (Alpharetta, GA)**

Impact Engineer of the Year, 2007–2008

- Sole backend developer for the Secure Web Proxy Service, a managed Web security/control system.
- Sole developer of the IronNet™ appliance, making use of techniques including latent semantic analysis, clustering, and Markovian discrimination to prevent data leakage. This included the *snares* ICAP server.
- Developed *garuda*, a CheckPoint® OPSEC module interfacing with the TrustedSource™ reputation service.
- Developed the *HURLBAT* protocol testing tool, and SMTP, HTTP and ICAP modules.
- PCT/US2008/051869. Detecting Image Spam. 2008-07-31.

US Patent

- **2000/08–2005/12: Senior Software Engineer, Reflex Security (Atlanta, GA)**

Co-founder

- Led research team, focused on intrusion prevention, parallelized and distributed intrusion detection, and multiple pattern matching. Implemented several techniques later published by academia. Led interviewing.
- Sole developer of code for the Reflex Interceptor (now Reflex IPS™ and Reflex MG™), a Layer-2 bridging NIPS running Linux. This included:
  - \* *tako*, an IPS application statefully analyzing multiple GigE links inline and in real time, performing a forwarding verdict on each frame via use of `mmap(2)` ed packet sockets + custom `netlink(7linux)` messages, and
  - \* *geso*, an SMTP proxy making use of a Kaspersky™ Anti-Virus backend to filter mail inline and in real time, designed to be trivially extended to other store-and-forward protocols.
- Assisted development of build systems, automated testing and benchmarking tools, and backend platform configuration management.
- Assisted development of kernel patches to expand the netlink socket infrastructure and filter on `mmap(2)`-backed sockets.
- PCT/US2004/023739. System and Method for Threat Detection and Response. 2005-02-03. US Patent

- **1998/12–1999/12: Teaching Assistant, GT College of Computing (Atlanta, GA)**

- Recitations with ~15 students, plus one-on-one meetings and grading of exams and homeworks. Classes included:
  - \* “Programming Language Principles” (CS 3411—Language design and comparative programming linguistics).
  - \* “Models and Translation” (CS 2330—Parsing, interpretation and compilation, virtual machines).
  - \* “Control and Concurrency” (CS 2430—Parallel computing, UNIX systems programming and ANSI/ISO C).
  - \* “Instruction Set Architecture” (CS 2760—Assembly language, stored programs, and architectural models).

- **1998/09–2000/02, 2004/01–2005/05: Georgia Institute of Technology, Bachelor of Science in Computer Science**

- Specializations in Theory, Systems, and Networking.
- 2000 ACM Programming Team, GT Team A (“Gold”) (Captain).
- 1999 ACM Programming Team, GT Team B (“White”) (Captain).
- Research under Professor Wenke Lee (Topics: IDS, botnet models, anti-bot tech).

## Awards and Distinctions

- 2x Technical Infrastructure Award, Google (2014, 2015).
- Senior Member of the ACM (2012).
- Credited corrections to Donald Knuth’s *The Art of Computer Programming* and George Varghese’s *Networking Algorithmics*.
- General Secretary, GTISC DNS Security Summit (2009).
- Impact Engineer of the Year, McAfee (2008).
- Faculty Honors, Georgia Tech (1999).
- Dean’s List, Georgia Tech (1998).
- Third place, 1998 Questions Unlimited! National Academic Bowl Championships (New Orleans, LA).
- Second place, 1997 Questions Unlimited! National Academic Bowl Championships (Chicago, IL).
- Winner, 2016 Trivial Dispute All-City Tournament of Champions (New York, NY).

## Skills

Algorithmic thinking. ANSI/ISO C. Development in the UNIX environment. Design and analysis of algorithms. Network programming. Effective use and internals of Linux, FreeBSD, and their associated libs and threading implementations, GCC, GNU binutils, LLVM and the open source toolchain. Robust system design. Network security. Intrusion prevention. Establishment and detection of covert channels. String algorithms. Applications of automata theory. Parallel algorithms and design for multicore/manycore machines. Design of programming languages. POSIX APIs along with Linux and FreeBSD extensions thereof. Analysis of binaries. Most major network protocols and their primary open source implementations. Effective use of cryptography. High-throughput, low-latency I/O models, scalable I/O and its implementation, zero-copy networking, and network hardware design. Tools of network enumeration and domination. x86 assembly language, including MMX/SSE/AVX. Computer architecture. Compiler design. Elegant system administration. Bayesian methods. Combinatorics, stochastics, topology, analysis, *ad nauseam*.

- 
- **Operating Systems:** Linux (22 years), FreeBSD (6 years), Solaris (3 years), Windows NT (3 years)
  - **Languages (expert—minimum 10 active years):** C, Bourne shell, x86 assembly, GNU Make
  - **Languages (professional—minimum 2 active years):** C++, Python, Go, XSLT/XPath
  - **Languages (amateur—minimum 1 serious project):** Numerous assembly languages, Java, Haskell, Rust, Prolog, Lisp, ML, Scheme, JavaScript (and XUL), Scala, Clojure
  - **Technologies:** GCC, LLVM, Pthreads, OpenMP, CUDA, OpenCL, Berkeley sockets, ACE, Flex/Bison, ANTLR, LaTeX, Apache, Postfix, MediaWiki, DJBDNS, nginx, Bugzilla, VPNs, SSH, DocBook, GDB, valgrind, strace, IDAPro, Gecko/XULRunner, nmap, OProfile, perf, tcpdump, Pin, DynamoRIO, GnuTLS, OpenSSL, Avahi, iptables, Git, svn, Portsnap, ipfw, LARTC, socat, KVM/QEMU, Xen, VMWare ESXi, BLAS, ATLAS... If it can be done on a Linux box, I’ve probably done it.
-