SRI International Computer Science Laboratory

Unless you came through the main csl.sri.com Web site, you might want to click on the above photo for an informal pool picture (which is already on the main site), taken by my wife, **Elizabeth S. Neumann**, or the more formal official SRI photo. For professional photos, please contact **Jim Sugar**, jimsugar@aol.com, 1-415-388-3344, fax 415-388-3345, 45 Midway Ave., Mill Valley, CA 94941, a former National Geographic photographer and all-around good guy.

Peter G. Neumann

CLICK HERE for informal photo==> [click for an informal picture]

CLICK HERE for a more formal photo==> [click here for a more formal picture]

Position:
    Principal Scientist
Address:
    Computer Science Laboratory
    333 Ravenswood Ave EL-243
    Menlo Park California 94025-3493, USA
E-mail: Neumann@csl.sri.com
Webpage: http://www.csl.sri.com/neumann
Tel: 650/859-2375
    (if you don't like voicemail, press "0" to speak with a real human being.)

Click here for a **short bio.** More detailed bio information is available on request.

This Web page (http://www.csl.sri.com/neumann) can also be reached from the primary CSL Web site (http://www.csl.sri.com) by clicking on "CSL Staff" and then "Neumann". (It differs from the default CSL page.) The following sections are included here, and can be moused directly if you do not want to read linearly.

EXHIBIT
J

Academic and R&D Background

I have been a member of the SRI International Computer Science Laboratory since September 1971. I spent eight years at Harvard (1950-58, with my A.B. in Math in 1954, S.M. in Applied Math in 1955, and PhD in 1961 after returning from my two-year Fulbright in Germany (1958-60), where I also received the German Dr rerum naturarum in 1960.

The work for my two doctoral theses (Tony Oettinger was my Harvard advisor, and Alwin Walther my Darmstadt advisor) and various subsequent papers involved variable-length Huffman-like codes and later was extended to Huffman-style information-lossless sequential coding schemes with surprisingly strong self-resynchronization properties despite arbitrary fault modes and denial-of-service attacks, even in the presence of very low or minimum redundancy as in Huffman codes. These schemes provided the possibility of highly survivable communication systems in the presence of arbitrary temporary interference. Earlier, my undergraduate thesis in mathematics (1954) involved identifying five nomographic classes of motions based on elliptic integrals, establishing canonical transformations for each of those classes, and generating tables for them (using the Harvard Mark IV).

I had two reverse sabbaticals as Visiting Mackay Lecturer, during the spring quarter of 1964 at Stanford University in Electrical Engineering, and the academic year 1970-71 at U.C. Berkeley (teaching courses in hardware, operating systems, and coding theory, and co-leading two seminar courses). I also taught a course on survivable systems and networks at the University of Maryland in the fall of 1999, half in person, half by video teleconference; the course notes are indicated below.

My first computer job was in the summer of 1953, as a programmer on the IBM Card-Programmed Calculator, for the U.S. Naval Ordnance Lab in White Oak MD, a punched-card machine with four registers and ZERO memory. (The cards provided auxiliary memory!) Among other things, I wrote a nifty recursive complex matrix-inversion routine. The three-address instruction interpretation was done in the plugboard, which represented an early assembler! My boss was Cal Elgot, who later became director of the IBM mathematics group at IBM in its very early days at the Lamb Estate, before the research effort moved to the Watson Lab in Yorktown Heights, NY.

I had ten exciting years in the Computer Science Lab at Bell Labs in Murray Hill, New Jersey (1960-70) -- including extensive involvement in Multics from 1965 to 1969. Beginning in 1965, Bob Daley (then at Project MAC at MIT) and I did the Multics file system design, which included directory hierarchies, access-control lists (ACLs), dynamic linking of symbolic names to cacheable descriptor-based addresses, and dynamically paged segments within a novel hardware-supported virtual memory concept. (It is nice to find dynamic linking again being ``rediscovered'' in Webware! Multics also had multiprogramming, multiprocessing, multiple protection domains, and other forms of multiplexing.) I had a minor role in the Multics input-output design, heavily influenced by Ken Thompson, Joe Ossanna, and Stan Dunten, with symbolic stream names (which Ken later transmogrified into Unix pipes) and device-independent I/O. After Vic Vyssotsky moved over to Whippany, I found myself the Bell Labs member of the Multics Triumvirate, coordinating with Fernando Corbató (Corby) at MIT and Charlie Clingen at Honeywell, and flying to MIT for a meeting almost every other week. There was some really beautiful innovation in Multics, and many wonderful people. For those of you who are young folks with little idea of Multics' contributions to computer history, check out Tom Van Vleck's Multicians website at http://www.multicians.org/, which (as of 19 Feb 2007) listed 1880 names of people who were associated with Multics! Particularly notable among those not already mentioned is Jerry Saltzer, although many others were important contributors as well.

Click here for a few selected bibliographic references and other items. A list of CSL-related .bib entries is available at the bottom of the official CSL Web site page for me .

Research Interests at SRI

My main research interests continue to involve security, crypto applications, overall system survivability, reliability, fault tolerance, safety, software-engineering methodology, systems in the large, applications of formal methods, and risk avoidance. (I am apparently an Eclectical Engineer, a Zennish ZScientist, and a Peregrine Philosopher. A profile on me in the February 1999 issue of ICSA's Information Security magazine in pdf and in PostScript depicts me as a *``designated holist''*.) A short article on Holistic Systems summarizes the challenges of developing trustworthy systems holistically, with possible lessons from energy, health care, and agriculture. (This appeared in the *ACM SIGSOFT Software Engineering Notes,* 31, 6, November 2006, pages 4--5.)

Trustworthiness: Security

My coauthors Matt Bishop, Sean Peisert, Marv Schaefer, and I wrote a paper, Reflections on the 30th Anniversary of the IEEE Symposium on Security and Privacy, for the May 2010 proceedings of the 31st annual meeting. We regret inadvertently omitting recognition of Sushil Jajodia for the most accepted papers (in Section VII) and Gerry Popek [d] (in Section IX). The paper is of course subject to IEEE copyright, but you have my permission to use it for educational and noncommerical purposes.

I gave a keynote talk, **Identity and Trust in Context,** for IDtrust 2009 at NIST on 15 April 2009. The slides are online at the conference website and on my website. This talk included discussion of the importance of holistic system considerations rather than trying to deal with identity and authorization in isolation, with applications to health care, and summarized the work of Brent Waters (Attribute-Based Encryption), Carl Gunter (Attribute-Based Messaging), and Chris Peikert (Lattice-Based Cryptography).

In the early 2000s, DARPA funded thirteen projects under its **Composable High-Assurance Trustworthy Systems (CHATS)** program, created by Douglas Maughan. I led one of those projects **(CHATS project website)**, in the SRI Computer Science Laboratory. The emphasis in the CHATS program was on composable trustworthy open-source operating systems. The final report, Principled Assuredly Trustworthy Composable Architectures, was completed on 28 December 2004, and is available in three forms: html, pdf, and ps. An earlier paper summarizing the project as of early 2003 appeared in the DISCEX03 proceedings: Achieving Principled Assuredly Trustworthy Composable Systems and Networks.

SRI's Computer Science Lab and the University of Cambridge have embarked on a new project for the DARPA CRASH program (Clean-slate design of Resilient, Adaptive, Survivable Hosts), which is called **CTSRD** (CRASH-worthy Trustworthy Systems R&D). The first paper resulting from this project, Peter G. Neumann and Robert N. M. Watson, Capabilities Revisited: A Holistic Approach to Bottom-to-Top Assurance of Trustworthy Systems, was presented at the Fourth Layered Assurance Workshop (in association with ACSAC 2010) in Austin Texas, 6-7 December 2010. A new paper reflects subsequent progress on the development of the hardware architecture, CHERI: A Research Platform Deconflating Hardware Virtualization and Protection for the RESoLVE workshop associated with ASPLOS in London, March 2012.

Incidentally, a significant effort is underway in Peter Denning's Great Principles project, which considers the importance of principles more broadly --- as common elements across system designs. He is in the process of writing a book on that effort.

The **Provably Secure Operating System (PSOS)** project began in 1973 and continued until 1983. The 1980 PSOS final report (noted in my partial reference list) has been scanned in and is online in PostScript form (over 300 pages). The report includes the system architecture and many of the basic hardware and operating system layers, plus some illustrative applications (all formal specified in the SPECIAL language of HDM, the Hierarchical Development Methodology). The Feiertag/Neumann paper summarizing the architecture as of 1979 is available in a retyped, more or less correct, hand-edited pdf form. A 2003 paper, PSOS Revisited by me and Rich Feiertag, was presented at ACSAC 2003 in Las Vegas in December 2003, as part of the Classic Papers track (which was initiated at ACSAC 2002 for the Karger-Schell paper on the Multics multilevel secure evaluation). Please read it if you are interested in capability architectures. The PSOS project continued from 1980 to 1983, supporting the Goguen-Meseguer papers and the Extended HDM effort that led to SRI's PVS system.

A 1996 report, **Architectures and Formal Representations for Secure Systems,** considers what formal methods can do for system security, and vice versa. It is available in PostScript form. and contains various references to earlier work, e.g., to our 1970s work on the formally specified capability-based object-oriented hierarchically-layered **Provably Secure Operating System (PSOS)**, and the role of system structure and abstraction -- which has been a long-standing interest. A 1992 paper by **Norm Proctor** and me, **Architectural Implications of Covert Channels**