Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Index to Volume 3

Sunday, 2 November 1986

Volume 3 Issue 1 (4 Jun 86)

- Unshakeable Faith in Technology (Richard A. Cowan)
- Unshakeable Faith in Technology: Shuttles & Nuclear Power (Peter G. Neumann)
- Basis for SDI Assumptions? (Doug Schuler)
- Technical vs. Political in SDI (Herb Lin)
- Computer Crime Laws (Peter G. Neumann)
- Backups for micros (Evan Dresel)
- The Clock Lies Again (PGN, Jagan Jagannathan)

Volume 3 Issue 2 (5 Jun 86)

- Are SDI Software predictions biased by old tactical software? (Herb Lin)

Volume 3 Issue 3 (6 Jun 86)

- Watch this Space (Eugene Miya)
- Unshakeable Faith in Technology (Herb Lin)
- SDI as a defense against terrorists? (Bruce Wampler, Martin Moore, Bernie Gunther)
- Basis for SDI Assumptions? (Herb Lin)

Volume 3 Issue 4 (9 Jun 86)

- Re: Watch this Space (Mark Jackson, Eugene Miya)
- Software developer's liability (Paul Schauble)
- What an Algorithm!! (Brian Bishop)
- Sgt. York's Latrine, and other stories (Mike McLaughlin, Ken Laws)

Volume 3 Issue 5 (10 Jun 86)

- A powerful metal detector and magnetic personalities with bank cards (Matthew P. Wiener)
- Shuttle Launch Decisions (Don Wegeng)
- Re: Estell's defense of SDI (Martin Purvis)
- Sgt. York's Latrine, and other stories (Mike McLaughlin)

Volume 3 Issue 6 (12 Jun 86)

- Risks from inappropriate scale of energy technologies (Michael J. Natkin)
- Shuttle Software (David C. Smith)

- An additional SDI problem: sensor technology (Eugene Miya)
- Privacy in the electronic age (Dave Platt)
- Sgt York software (Larry Campbell, Mark Vilain)

Volume 3 Issue 7 (13 Jun 86)

- Eastport Study Group report ("Science" article) (Pete Kaiser)
- An additional SDI problem: sensor technology (Jon Jacky)
- Shuttle software and CACM (James Tomayko [and Herb Lin])
- Privacy laws (Bruce O'Neel)
- A mini-editorial on running the RISKS Forum (PGN)

Volume 3 Issue 8 (15 Jun 86)

- Challenger, SDI, and management risks (Dick Dunn)
- Re: Risks from inappropriate scale of energy technologies (Chuck Ferguson)
- Distributed versus centralized computer systems (Peter G. Neumann)
- Privacy legislation (Michael Wagner)

Volume 3 Issue 9 (20 Jun 86)

- Informing the Senate on SDI (Jim Horning)
- A medical risk of computers (Karen R. Sollins)
- Risks of VDTs (Alan Wexelblat)
- Minor addition on Risks of Distributed Energy (Ted Lee)

Volume 3 Issue 10 (20 Jun 86)

- Re: Privacy Legislation & Cellular Swiss Cheese (RISKS-3.8)(Geoff Goodfellow)
- Re: Privacy Legislation (RISKS-3.6) [divulging] (Dan Franklin)
- Re: Privacy Legislation (RISKS-3.6) [radar detectors] (Herb Lin)

Volume 3 Issue 11 (23 Jun 86 [mislabelled RISKS-3.12 in masthead])

- A medical risk of computers (overdose during radiation therapy) (Jon Jacky)
- Secure computer systems (Herb Lin)
- Radar Detectors (Re: Privacy legislation in RISKS-3.10) (Jeff Makey)
- Telco Central office woes in Southfield, MI. (via Geoff Goodfellow)
- Reducing the managerial risks in SDI (Bob Estell)
- Economic Impact of SDI: Transcript Info (Richard A. Cowan)

Volume 3 Issue 12 (24 Jun 86)

- License Plate Risks (Chuck Price)
- SDI is for ICBMs, Not Terrorists (Mark Day)
- Still another kind of clock problem (Rodney Hoffman)
- Estimating Unreported Incidents (Ken Laws)
- Estimating Unreported Incidents -- and the risks of using statistics (PGN)
- Re: Privacy legislation (RISKS-3.8) and radio eavesdropping (Jerry Mungle, Jeff Mogul, Jim Aspnes)

Volume 3 Issue 13 (26 Jun 86)

- The Risky Gap Between Two Design Cultures (Jack Goldberg)
- Risks of nuclear power (Dan Franklin)
- Research programs that pay for themselves (Rich Cowan)
- Having an influence from "within the system" (Rich Cowan)
- RISKS in running RISKS -- continued (PGN and an unhappy Mailer)

- Volume 3 Issue 14 (27 Jun 86)
  - A Personal View on SDI (Harlan Mills)
  - Privacy legislation (RISKS-3.10) (Jerome H. Saltzer)
  - Risks in burning wood (Mike McLaughlin)
  - Mailer explosion (Sean Malloy)
- Volume 3 Issue 15 (29 Jun 86)
  - A Personal View on SDI from Harlan Mills (Herb Lin)
  - Having an influence from "within the system" (Herb Lin)
  - Re: Research programs that pay for themselves (Rich Cowan)
  - Text Scanners (Fred Hapgood)
- Volume 3 Issue 16 (30 Jun 86)
  - Chernobyl (a suprise to the Soviets) (Martin Minow)
  - Airwaves & Security (2 Subjects) (Richard S. D'Ippolito via dhm)
  - Interesting Technical Questions (originally SDI) (Martin Moore)
- Volume 3 Issue 17 (3 Jul 86)
  - How Much Computer Literacy Is Enough? (JAN Lee)
  - Working within the system (Rich Cowan)
  - Re: [Airwaves &] Security -- SDI (Herb Lin)
  - Complex issues, complex answers (Bob Estell)
  - Politics and Engineering Practice (Seifert)
  - Multiple copies of RISKS-3.16 (Kenneth Sloan)
  - GTE Sprint billing problems (Chuck Weinstock/Lee Breisacher)
- Volume 3 Issue 18 (8 Jul 86)
  - Computer Crime in Scandinavia (Martin Minow)
  - Re: Risks from inappropriate scale of energy technologies (Henry Spencer)
  - Sensor technology and disinformation (Eugene Miya)
  - Educating to prevent RISKS (Steven Gutfreund)
  - Rash of 'Undeliverable mail' (Chuck Price)
- Volume 3 Issue 19 (10 Jul 86)
  - Computer Literacy (Rick Smith, Bob Estell, Col. G. L. Sicherman, PGN)
- Volume 3 Issue 20 (15 Jul 86)
  - Risks of computer incompetence (Dave Benson)
  - RE: educating about RISKS (Don Lindsay)
  - Computer Literacy (RISKS-3.19) (Ron Morgan) ... and Basic (Martin Minow, Andrew Klossner, PGN)
  - Dial-up computing (Sterling Bjorndahl)
  - Research programs that pay for themselves (Clayton Cramer)
- Volume 3 Issue 21 (16 Jul 86)
  - Responsibility (Willis Ware)
  - Programming languages and computer literacy (Bob Estell)
  - Teaching about risks, BASIC, NASA, etc. (Eugene Miya)
  - Programming Languages (Matthew Kruk)
  - BBoard Lingo (Trojan viruses,...) (Hank Burchard, via Peter G. Neumann)

- Volume 3 Issue 22 (19 Jul 86)
  - Nostalgia (Mike Williams)
  - Flames about BASIC (Jim Anderson)
  - More on risks of teaching "just" programming (Herb Lin)
  - Responsibility for Computer Actions (George S. Cole)
  - CDP and Certification (Andy Glew)
  - The undetected hang-up risk (more) (Ted Lee)
- Volume 3 Issue 23 (22 Jul 86)
  - Re: Comet and Electra (Jim Horning)
  - 100,000 Late Phone Bills (Mike McLaughlin)
  - Types of "Programming" (Henry Schaffer)
- Volume 3 Issue 24 (24 Jul 86)
  - Comet and Electra (Jerry Saltzer, Marv Zelkowitz, Don Chiasson, Bard Bloom)
  - No gasoline because the computer is down? (Jim Barnes)
  - HBO Hacker Captain Midnight Caught (via Geoff Goodfellow)
- Volume 3 Issue 25 (24 Jul 86)
  - Petroski on the Comet failures (Alan Wexelblat)
  - Re: Comet and Electra (Douglas Adams)
  - On the dangers of human error (Brian Randell via Lindsay Marshall)
  - Software Paranoia (Ken Laws)
  - Royal Wedding Risks (Lindsay Marshall)
  - How to Think Creatively (John Mackin)
  - Dangers of improperly protected equipment (Kevin Belles)
- Volume 3 Issue 26 (26 Jul 86)
  - DIVAD (Herb Lin)
  - Royal wedding risks -- common change modes (Don Chiasson)
  - Security and dialbacks (David I. Emery via Herb Lin) [Long message]
- Volume 3 Issue 27 (29 Jul 86)
  - Whoops! Lost an Area Code! (Clayton Cramer)
  - Comet-Electra (RISKS-3.25) (Stephen Little)
  - Comparing computer security with human security (Bob Estell)
- Volume 3 Issue 28 (31 Jul 86)
  - Laserprinter dangers (Mansfiel)
  - Errors in error-handlers (Mansfiel)
  - Military testing errors (Alan Wexelblat)
  - Re: Comet-Electra (RISKS-3.25) (Bill Fisher)
  - Computer and Human Security (Lindsay Marshall)
- Volume 3 Issue 29 (1 Aug 86)
  - Ozone hole undetected for years due to programming error (Bill McGarry)
  - Aircraft simulators and risks (Art Evans)
  - Military testing errors (Scott E. Preece)
  - Risks: computers in the electoral process (Kurt Hyde via Pete Kaiser)
  - Risks of CAD (Alan Wexelblat)

- Volume 3 Issue 30 (4 Aug 86)
  - Ozone hole undetected (Jeffrey Mogul)
  - Re: Risks of CAD (Henry Spencer)
  - Comment on Hartford Civic Roof Design (Richard S D'Ippolito)
  - Expert system to catch spies (Larry Van Sickle)
- Volume 3 Issue 31 (5 Aug 86)
  - Another cruise missile lands outside Eglin test range (Martin J. Moore)
  - Aircraft simulators and risks (Gary Wemmerus)
  - Re: Comment on Hartford Civic Roof Design (Brad Davis)
  - Expert system to catch spies (RISKS-3.30) (Chris McDonald)
  - Computer and Human Security (Henry Spencer)
  - Ozone Reference (Eugene Miya)
  - Financial risks (Robert Stroud)
  - Mail Load Light(e)ning? (SRI-CSL Mail Daemon)
- Volume 3 Issue 32 (6 Aug 86)
  - DC-10 Crash (Chuck Weinstock)
  - Earthquake Reporting (AP)
  - The Recent Near-Disaster for the Shuttle Columbia (Peter G. Neumann)
  - Traffic lights in Austin (Alan Wexelblat)
  - Re: Laserprinter dangers (Graeme Hirst)
- Volume 3 Issue 33 (7 Aug 86)
  - Air traffic computer failure (Hal Perkins)
  - Re: Laserprinter dangers (Sean Malloy)
  - Re: Expert system to catch spies (Rich Kulawiec)
  - Survey of Computer Professionals (Kurt Hyde)
- Volume 3 Issue 34 (9 Aug 86)
  - Non-Flying Airplanes and Flying Glass (Jim Horning)
  - Failure Recovery, Simulations, and Reality (Danny Cohen)
  - Ottawa Power Failure (Dan Craigen)
  - Liability for Software Problems (Peter G. Neumann)
  - Ozone hole (Hal Perkins)
  - Re: Survey of Trust in Election Computers (Chris Hibbert)
  - Nondelivery of RISKS-2.38 (8 April 1986) and other mail (Communications Satellite [and PGN])
- Volume 3 Issue 35 (11 Aug 86)
  - Flying windows on the Hancock Building (Remy Malan)
  - Pilots and counter-intuitive maneuvers (Martin Minow)
  - Mail adrift (Mike McLaughlin)
  - Laserprinter dangers (Niall Mansfield)
  - A bit of humor and even philosophy (Willis Ware)
  - Official Report on Chernobyl disaster (Robert Stroud)
- Volume 3 Issue 36 (12 Aug 86)
  - Another Medical Risk? (Lee Breisacher)
  - RISKy Business in Surgery (Mark Jackson)

Reliance on word-processors discussed in the Israeli Supreme (Ady Wiernik)

- Expert Systems - The New Cop on the Beat (Laws via Fred Ostapik)
- Chernobyl (Art Evans, Dick Karpinski)
- Air Traffic Control computer failure (Dan Melson)
- Possible failures of BMD software (Herb Lin)
- A note about stories "from memory" (Henry Mensch)

Volume 3 Issue 37 (14 Aug 86)

- Computer Viruses (Robert Stroud)
- On knowing how hard a system is to make work (Bob Estell)
- COMSAT and the Nondelivery of Mail (Rob Austein)
- Exploding Office Chairs (Jonathan Bowen)

Volume 3 Issue 38 (17 Aug 86)

- Computer gives away California state funds (Rodney Hoffman)
- High-Tech Sex Ring: Beware of Whose Database You Are In! (Peter G. Neumann)
- Computer Viruses (Chris McDonald, Paul Garnet, Matt Bishop)
- Computer Viruses and Air Traffic Control (Dan Melson)
- Re: Traffic lights in Austin (Bill Davidsen)

Volume 3 Issue 39 (19 Aug 86)

- Nuclear false alarm (Robert Stroud)
- Risk to beer production? (Robert Stroud)
- Re: High Tech Sex (Lindsay F. Marshall)
- QA on nuclear power plants and the shuttle (Roy Smith)
- Hackers in BITNET (Sterling Bjorndas)

Volume 3 Issue 40 (21 Aug 86)

- QA on nuclear power plants and the shuttle (Eugene Miya, Ken Dymond)
- CAD, Simulation, Armored Combat Earthmover, and Stinger (Mary C. Akers)
- Risks Distribution List -- Private-Copy Subscribers PLEASE READ! (PGN)
- Could computers launch a nuclear attack? (Jeff Myers)

Volume 3 Issue 41 (23 Aug 86)

- $1 million bogus bank deposit (Hal Perkins)
- Cheating of automatic teller machines (Jacob Palme)
- Simulation, Armored Combat Earthmover, and Stinger (Herb Lin)
- Report from AAAI-86 (Alan Wexelblat)

Volume 3 Issue 42 (25 Aug 86)

- Re: $1 million bogus bank deposit (Barry Shein)
- Sometimes things go right (Matt Bishop)
- Re: Cheating of automatic teller machines (Dave Farber)
- Keystroke Analysis for Authentication (rclex)
- Computer Vote Counting In the News -- More (John Woods)

Volume 3 Issue 43 (26 Aug 86)

- Comment on PGN's comment on human error (Nancy Leveson)
- Keystroke Analysis for Authentication (Scott E. Preece, Eugene Miya)
- Risks of Mechanical Engineering [More on O-Rings] (Martin Harriman)

Re: Words, words, words... (Mike McLaughlin)
- Comments on paper desired (Herb Lin)

- Volume 3 Issue 44 (27 Aug 86)
  - F-16 Problems (George Moore via Bill Janssen)
  - Various clips from European Newspapers (Martin Minow)
  - Comment on Nancy Leveson's comment on... (Alan Wexelblat)
  - Words, words, words... (Herb Lin)
  - Software Safety (Paul Anderson)
- Volume 3 Issue 45 (28 Aug 86)
  - Nonviolent Resistor Destroys Aries Launch (PGN)
  - Risks in the design of civil engineering projects (Annette Bauman)
  - ATMs (Lindsay F. Marshall)
  - Re: Typing Profiles (Lindsay F. Marshall)
  - Human errors prevail (Ken Dymond, Nancy Leveson)
- Volume 3 Issue 46 (30 Aug 86)
  - Human error (Nancy Leveson, Lindsay F. Marshall)
  - Re: F-16 Tales (Earl Boebert, Phil Ngai)
  - Correction to note about flight simulators (Martin Minow)
  - Supermarket grinds to a halt (David Sherman)
  - Video processing (Guy Schafer)
  - ATMs (Jacob Palme)
- Volume 3 Issue 47 (1 Sep 86)
  - Flight Simulators Have Faults (Dave Benson)
  - Re: QA on nuclear power plants, the shuttle, and beer (Henry Spencer)
  - Acts of God vs. Acts of Man (Nancy Leveson -- two messages)
  - Computer Literacy (Mike McLaughlin)
  - Another supermarket crash (Ted Lee)
  - A supermarket does not grind to a halt (Brint Cooper)
- Volume 3 Issue 48 (2 Sep 86)
  - Aeromexico Crash (UPI via PGN)
  - Air Force puts secrets up for sale (Peter G. Neumann)
  - Randi, Popoff, and Data Privacy Laws (Phil Karn via Geoff Goodfellow)
  - Flight Simulators Have Faults (Gary Whisenhunt)
  - On-Line with Taco Bell Telephone (John Mulhollen)
  - Titanic photo expedition (Lindsay F. Marshall)
  - New Zealand $1 million deposit (Dave Sherman)
  - Examination Processing Error (Joe Stoy)
- Volume 3 Issue 49 (4 Sep 86)
  - Human Error (Dave Parnas, Bill Anderson)
  - Machine errors - another point of view (Bob Estell)
  - Flight simulators (Eugene Miya)
  - F-16 software (Henry Spencer)
  - Terminal (!) lockup (Ken Steiglitz)
- Volume 3 Issue 50 (7 Sep 86)

- Enlightened Traffic Management (Alan Wexelblat)
- Flight Simulator Simulators Have Faults (Dave Benson)
- Re: Flight Simulators and Software Bugs (Bjorn Freeman-Benson)
- Always Mount a Scratch Monkey (Art Evans)
- Re: supermarket crashes (Jeffrey Mogul)
- Machine errors - another point of view (Bob Estell)
- Human Behv. & FSM's (Robert DiCamillo)

Volume 3 Issue 51 (7 Sep 86)

- Computer almost created swing vote (Bjorn Freeman-Benson)
- Computer Sabotage of Encyclopedia Brittania (Rosanna Lee)
- F-16 software (Wayne Throop)
- Arbiter failures and design failures (Martin Harriman)
- Systems errors (hardware AND humans) (Bill Janssen)
- Re: Terminal (!) lockup (Roy Smith)

Volume 3 Issue 52 (8 Sep 86)

- Re: F-16 software (Nancy Leveson)
- Upside-down F-16's and "Human error" (Jon Jacky)
- F-16 software (Scott E. Preece)
- Do More Faults Mean More Faults? (Ken Dymond)
- Why components DON'T interact more often (Bob Estell)
- Computer almost created swing vote (Scott E. Preece)
- Computer Sabotage [MISSING LAST LINE FROM RISKS-3.51]
- Computer Sabotage of Encyclopedia Brittanica (Scott E. Preece)
- Captain Midnight & military satellites (Werner Uhrig)
- Re: always mount a scratch monkey (Alexander Dupuy)
- Erroneous computer printout used in public debates (Chris Koenigsberg)

Volume 3 Issue 53 (10 Sep 86)

- Hardware/software interface and risks (Mike Brown)
- More on Upside down F-16s (Mike Brown)
- "Unreasonable behavior" and software (Gary Chapman)
- Re: supermarket crashes (Scott Preece)

Volume 3 Issue 54 (15 Sep 86)

- Ada Inherently Secure? (Mike McLaughlin)
- A million lines of code works the first time? (Ken Calvert)
- Computers and Ethics (Mark S. Day)
- New book: HUMAN RELIABILITY: With Human Factors (Elizabeth ?)
- Answers to WWMCCS Intercomputer Network questions (Harold E. Russell)

Volume 3 Issue 55 (15 Sep 86)

- Hardware/software interface and risks (Kevin Kenny)
- F-16 (Holleran, Eugene Miya, Ihor Kinal, Doug Wade)

Volume 3 Issue 56 (16 Sep 86)

- Massive UNIX breakins at Stanford (Brian Reid)

Volume 3 Issue 57 (16 Sep 86)

- Computers and the Stock Market (again) (Robert Stroud)
- The Old Saw about Computers and TMI (Ken Dymond)
- Do More Faults Mean (Yet) More Faults? (Dave Benson)
- A critical real-time application worked the first time (Dave Benson)
- Autonomous weapons (Eugene Miya)
- "Unreasonable behavior" and software (Eugene Miya on Gary Chapman)
- Risks of maintaining computer timestamps revisited (John Coughlin)

Volume 3 Issue 58 (17 Sep 86)

- Massive UNIX breakins (Dave Curry, Brian Reid)
- "Atlanta's been down all afternoon" (Alan Wexelblat)
- F-16 software (Herb Lin)
- Viking Project (Eugene Miya)
- Protection of personal information (David Chase)
- Autonomous Weapons (Ken Laws)
- Re: computers and petty fraud (Col. G. L. Sicherman)

Volume 3 Issue 59 (20 Sep 86)

- Computers and Wall Street (Robert Stroud)
- Report from the Computerized Voting Symposium (Kurt Hyde)
- Computers, TMI, Chernobyl, and professional licensing (Martin Harriman)
- Failsafe software (Martin Ewing)
- Software vs. Mechanical Interlocks (Andy Freeman)
- How Not to Protect Communications (Geoff Goodfellow)

Volume 3 Issue 60 (20 Sep 86)

- Sanity checks (Roy Smith)
- Viking Flight Software working the `first' time? (Greg Earle)
- A million lines of code works the first time? (Anonymous, Dave Benson, Herb Lin)
- Re: Massive UNIX breakins at Stanford (Scott E. Preece)
- Re: Protection of personal information (Andy Mondore, Herb Lin)
- Announcement of Berkeley Conference on the SDI (Eric Roberts)

Volume 3 Issue 61 (21 Sep 86)

- Computers and Ethics (Robert Reed)
- Autonomous weapons (Wayne Throop)
- Simulation risk (Rob Horn)
- Viking software (James Tomayko)
- Risks of passwords on networks (Bruce)
- More on digital jets; Sanity checks (Eugene Miya)

Volume 3 Issue 62 (22 Sep 86)

- Massive UNIX breakins at Stanford (Jerry Saltzer, Rob Austein, Andy Freeman, Scott Preece)
- F-16 Software (Henry Spencer)
- 1,000,000 lines of correct code? (Stephen Schaefer)

Volume 3 Issue 63 (24 Sep 86)

- NOTROJ (a Trojan Horse) (James H. Coombs via Martin Minow)
- Massive UNIX breakins at Stanford (Scott Preece [two more messages!])

Volume 3 Issue 64 (24 Sep 86)

- Sane sanity checks / risking public discussion (Jim Purtilo)
- More (Maybe Too Much) On More Faults (Ken Dymond)
- Re: Protection of personal information (Correction from David Chase)
- Towards an effective definition of "autonomous" weapons (Herb Lin, Clifford Johnson [twice each])

Volume 3 Issue 65 (24 Sep 86)

- UNIX and network security again (Andy Freeman)
- F-16 software (Wayne Throop)
- NYT feature article on SDI software (Hal Perkins)
- Autonomous widgets (Mike McLaughlin)
- Robottle Management Software? (PGN)

Volume 3 Issue 66 (25 Sep 86)

- Follow-up on Stanford breakins: PLEASE LISTEN THIS TIME! (Brian Reid)
- F-16 software [concluded?] (Herb Lin)

Volume 3 Issue 67 (25 Sep 86)

- Old GAO Report on Medical Device Software (Chuck Youman)
- Re: Stanford breakin, RISKS-3.62 DIGEST (Darrel VanBuer)
- Re: Passwords and the Stanford break-in (RISKS-3.61) (Dave Sherman)
- Re: role of simulation - combat simulation for sale (Jon Jacky)
- MIT Symposium on economic impact of military spending (Richard Cowan)
- "Friendly" missiles and computer error -- more on the Exocet (Rob MacLachlan)

Volume 3 Issue 68 (26 Sep 86)

- VDU risks -- Government changes its mind, perhaps (Stephen Page)
- "Drive by wire" systems (Charles R. Fry)
- Viking Landers worked the first time and met the specs (Dave Benson)
- Unix breakins - secure networks (David C. Stewart)
- Comment on the reaction to Brian's Breakin Tale (Dave Taylor)
- Reliability, complexity, and confidence in SDI software (Bob Estell)

Volume 3 Issue 69 (28 Sep 86)

- Confidence in software via fault expectations (Dave Benson)
- More on Stanford's UNIX breakins (John Shore, Scott Preece)
- F-16 simulator (Stev Knowles)
- Deliberate overrides? (Herb Lin)
- Viking Landers -- correction to RISKS-3.68 (Courtenay Footman)

Volume 3 Issue 70 (29 Sep 86)

- Deliberate overrides? (Scott E. Preece)
- Multiple causes and where to place the "blame" (PGN)
- The Art of "Science" and its Computers (PGN)
- No-lock Brakes (Peter Ladkin)
- Sanity in Automating Keyword Abstracting (Brint Cooper)
- The Network Is Getting Old? (PGN)

Volume 3 Issue 71 (30 Sep 86)

- Deliberate overrides? (Herb Lin, Alan M. Marcum, Eugene Miya)
- "Friendly" missiles and computer error - more on the Exocet (Robert Stroud)

- Re: Reliability, complexity, and confidence in SDI (Michal Young)
- My understanding of "path" and "bathtub curve" (Bob Estell)
- More artificial than intelligent? (Autokeywords) (Bob Estell)
- A Viking lander query (PGN)
- Note on ARPANET congestion (Nancy Cassidy)
- Indeed, the network is getting old (Jonathan Young)

Volume 3 Issue 72 (1 Oct 86)

- Viking Lander (Nancy Leveson)
- Deliberate override (George Adams)
- Overriding overrides (Peter Ladkin)
- A propos landing gear (Peter Ladkin)
- Paths in Testing (Mark S. Day)
- Confidence in software via fault expectations (Darrel VanBuer)

Volume 3 Issue 73 (2 Oct 86)

- Lessons from Viking Lander software (Bob Estell)
- Software wears out? (Rob Austein)
- Wrongful eviction through computer error (Bill Janssen)
- Deliberate override (Herb Lin, Ray Chen)
- Re: Piper Arrow Gear Override (Douglas Adams)
- Undesirable breakins and causes (Ian Davis)

Volume 3 Issue 74 (3 Oct 86)

- Opinions vs. Facts in RISKS Reports (re Aviation Accidents) (Danny Cohen)
- Mathematical checking of programs (quoting Tony Hoare) (Niall Mansfield)
- Risks of maintaining computer timestamps revisited [RISKS-3.57] (Ian Davis)
- Keyword indexing in automated catalogs (Betsy Hanes Perry)
- Re: Viking Landers -- correction (Scott Preece)
- Re: Confidence in software via fault expectations (Scott Preece)
- Overrides and tradeoffs (Jerry Leichter)
- Re: Deliberate overrides (Brint Cooper)
- Re: idiot-proof cars (risks-3.68) (Col. G. L. Sicherman)

Volume 3 Issue 75 (4 Oct 86)

- re: Estell on Viking (RISKS-3.73) (David Parnas, Dave Benson)
- Software becomes obsolete, but does not wear out (Dave Benson)
- The fallacy of independence (Dave Benson)
- Re: Paths in Testing (RISKS-3:72) (Chuck Youman, Mark Day)
- Mathematical checking of programs (quoting Tony Hoare) (Henry Spencer)

Volume 3 Issue 76 (5 Oct 86)

- Obsolescence vs wearing out (RISKS-3.75) (Jerome H. Saltzer)
- Cars, computers and unexpected interactions (Mike McLaughlin)
- Re: Mathematical checking of programs (quoting Tony Hoare) (Matthew Wiener)
- "Total correctness", "complete reliability" (RISKS-3.75) (Bard Bloom)

Volume 3 Issue 77 (8 Oct 86)

- Evaluating software risks (Brian Randell)
- Misapplication of hardware reliability models (Nancy Leveson)
- Deliberate overrides? (Mark Brader, Ephraim)

- Trusting-infallible-machines Stonehenge anecdote (Mark Brader)
- [More Aviation Hearsay?] (C Lewis)

Volume 3 Issue 78 (9 Oct 86)

- On models, methods, and results (Bob Estell)
- Fault tolerance vs. verification experiments (Nancy Leveson)
- The second Tomahawk failure (PGNeumann)
- Re: Overrides and tradeoffs (Eugene Miya, Herb Lin)
- Software getting old (Ady Wiernik)
- Rebuttal -- Software CAN Wear Out! (George Cole)
- "Obsolescence" and "wearing out" as software terms (Dave Benson)
- Obsolesence and maintenance - interesting non-software anecdote (Jon Jacky)
- FAA - Plans to replace unused computers with new ones (McCullough)

Volume 3 Issue 79 (12 Oct 86)

- China Air incident... the real story (Peter G. Trei)
- Air-Traffic Control Spoof (Peter G. Neumann)
- Aviation Accidents and Following Procedures (RISKS-3.77) (Matthew Waugh)
- DC-9 crash again (Peter Ladkin)

Volume 3 Issue 80 (15 Oct 86)

- US Navy reactors (Henry Spencer)
- Data Protection Act Risks (Lindsay F. Marshall)
- Is Bours(e)in on the Menu? (Martin Minow)
- Re: Software Wears Out (anonymous)

Volume 3 Issue 81 (19 Oct 86)

- System effectiveness is NOT a constant! (anonymous)
- Aircraft self-awareness (Scott Preece)
- Re: US Navy reactors (Brint Cooper, Eugene Miya, Stephen C Woods)
- Editorial on SDI (Michael L. Scott)

Volume 3 Issue 82 (20 Oct 86)

- NASDAQ computer crashes (Jerry Leichter, Vint Cerf)
- Sensors on aircraft (Art Evans, Henry Spencer)
- Loss of the USS Thresher (John Allred)
- Re: US Navy reactors (Henry Spencer)
- Risks from Expert Articles (Andy Freeman)

Volume 3 Issue 83 (21 Oct 86)

- Risks from Expert Articles (David Parnas, Herb Lin, Andy Freeman)
- Loss of Nuclear Submarine Scorpion (Donald W. Coley)
- Staffing Nuclear Submarines (Martin Minow)
- An SDI Debate from the Past (Ken Dymond)
- System effectiveness is non-linear (Dave Benson)
- Stealth vs Air Traffic Control (Schuster via Herb Lin)
- Missing engines & volcano alarms (Martin Ewing)

Volume 3 Issue 84 (22 Oct 86)

- Risks of using an automatic dialer (Bill Keefe)

Re: Missing engines & volcano alarms (Eugene Miya)

- False premise ==> untrustworthy conclusions (Martin Harriman)
- USN Automated Reactors (Dan C Duval)
- Keep It Simple as applied to commercial nuclear power generation (Martin Harriman)
- Works as Documented (Martin Minow)
- Re: Editorial on SDI (Michael L. Scott)
- Risks from Expert Articles (Herb Lin)
- Stealth vs. ATC / SDI Impossibility? / Missing Engines ? (Douglas Humphrey)

Volume 3 Issue 85 (23 Oct 86)

- On the Risk of Discussing SDI (Craig Milo Rogers)
- SDI Impossibility (Douglas Humphrey)
- Swedish Vulnerability Board Report on Complex System Vulnerabilities (Chuck Youman)
- Re: Thresher (David Feldman)
- Stealth and ATC (Dan Melson)
- Inoperative components (Peter Ladkin)

Volume 3 Issue 86 (26 Oct 86)

- Addition to Census of Uncensored Sensors (PGN)
- Military vs. civilian automatic control systems (Will Martin)
- Re: System effectiveness is non-linear (Scott E. Preece)
- SDI assumptions (Daniel M. Frank)
- SDI impossibility (David Chase)
- Editorial on SDI (Henry Spencer plus quote from David Parnas)

Volume 3 Issue 87 (26 Oct 86)

- System Overload (Mike McLaughlin)
- Information Overload (Mike McLaughlin)
- SDI assumptions (Herb Lin)

Volume 3 Issue 88 (27 Oct 86)

- SDI, Missing engines, feeping creatureism in consumer products (Roy Smith)
- More aircraft instrumentation (John Allred)
- Re: Military vs. civilian automatic control systems (Eugene Miya)
- Perfection (Douglas Humphrey)
- Shipboard anecdotes (Mike McLaughlin)
- RISKS UNDIGESTIFIER on UNIX (John Romine)

Volume 3 Issue 89 (28 Oct 86)

- Airplanes and risks (Alan Wexelblat)
- TSE, Air Canada (Matthew Kruk)
- Big Bang (Robert Stroud)
- Physicists on SDI and engineering.. (Herb Lin)
- ABM, SDI, and Freeman Dyson (Peter Denning)

Volume 3 Issue 90 (30 Oct 86)

- Anti Skid Brakes (Paul Schauble)
- The Mother's Day Myth, and "Old Reliable" (Jerome H. Saltzer)
- Collision avoidance systems (John Larson)
- Crime and punishment (Peter Ladkin)
- Air Canada (Matthew Kruk)

- (Voting) Machine Politics (Mike McLaughlin)
- Computer RISKS in "Ticker-Tape Parades" (PGN)
- SDI vs. Social Security (Scott Guthery)
- SDI Impossibility? (Scott Dorsey)
- Feeping Creaturism (Charley Wingate)

Volume 3 Issue 91 (30 Oct 86)

- Evolution, Progress (Jim Horning)
- System Overload (David Parnas)
- "Perfect" systems from imperfect parts (Bob Estell)
- The software that worked too well (Dave Benson)
- Assessing system effectiveness (Dave Benson)
- Risks of raining computer print-out (Alan Wexelblat, Martin Ewing, PGN)

Search RISKS using swish-e

Report problems with the web pages to the maintainer

THE RISKS DIGEST

Forum On Risks To The Public In Computers And Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Search RISKS using swish-e

The RISKS Forum is a moderated digest. Its USENET equivalent is comp.risks. (Google archive)

- Vol 26 Issue 47 (Monday 6 June 2011) <= Latest Issue
- Vol 26 Issue 46 (Saturday 4 June 2011)
- Vol 26 Issue 45 (Tuesday 24 May 2011)

- News about the RISKS web pages
- Subscriptions, contributions and archives

Feeds

RSS 1.0 (full text)

RSS 2.0 (full text)

ATOM (full text)

RDF feed

WAP (latest issue)

Simplified (latest issue)

Smartphone (latest issue)
Under Development!!

You can also monitor RISKS at Freshnews, Daily Rotation and probably other places too.

Please report any website or feed problems you find to the website maintainer. Report issues with the digest content to the moderator.

Selectors for locating a particular issue from a volume

Volume number: Volume Issue Number: Issue

Get Specific Issue

Volume Index

The dates and counts do not include the index issues for each volume.

Index to the RISKS Digest

Volume Number	Date Range	Number of Issues
Volume 1	*1 Aug 1985 - 31 Jan 1986*	*45 issues*
Volume 2	*1 Feb 1986 - 30 May 1986*	*56 issues*
Volume 3	*4 Jun 1986 - 30 Oct 1986*	*91 issues*
Volume 4	*2 Nov 1986 - 6 Jun 1987*	*96 issues*
Volume 5	*7 Jun 1987 - 31 Dec 1987*	*84 issues*

Volume 6	2 Jan 1988 - 31 May 1988	94 issues
Volume 7	1 Jun 1988 - 22 Dec 1988	98 issues
Volume 8	4 Jan 1989 - 29 Jun 1989	87 issues
Volume 9	6 Jul 1989 - 30 May 1990	97 issues
Volume 10	1 Jun 1990 - 31 Jan 1991	85 issues
Volume 11	4 Feb 1991 - 28 Jun 1991	95 issues
Volume 12	1 Jul 1991 - 24 Dec 1991	71 issues
Volume 13	6 Jan 1992 - 2 Nov 1992	89 issues
Volume 14	4 Nov 1992 - 27 Aug 1993	89 issues
Volume 15	2 Sep 1993 - 29 Apr 1994	81 issues
Volume 16	2 May 1994 - 22 Mar 1995	96 issues
Volume 17	27 Mar 1995 - 1 Apr 1996	96 issues
Volume 18	5 Apr 1996 - 31 Mar 1997	96 issues
Volume 19	1 Apr 1997 - 23 Sep 1998	97 issues
Volume 20	1 Oct 1998 - 31 Jul 2000	98 issues
Volume 21	15 Aug 2000 - 29 Mar 2002	98 issues
Volume 22	1 Apr 2002 - 27 Oct 2003	98 issues
Volume 23	7 Nov 2003 - 2 Aug 2005	96 issues
Volume 24	10 Aug 2005 - 30 Dec 2007	93 issues
Volume 25	7 Jan 2008 - 1 Apr 2010	98 issues
Volume 26	8 Apr 2010 - 6 Jun 2011	47 issues

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 1

Wednesday, 4 June 1986

Contents

- Unshakeable Faith in Technology
  Richard A. Cowan
- Unshakeable Faith in Technology: Shuttles & Nuclear Power
  Peter G. Neumann
- Basis for SDI Assumptions?
  Doug Schuler
- Technical vs. Political in SDI
  Herb Lin
- Computer Crime Laws
  Peter G. Neumann
- Backups for micros
  Evan Dresel
- The Clock Lies Again
  PGN
  Jagan Jagannathan
- Info on RISKS (comp.risks)

Unshakeable Faith in Technology

Richard A. Cowan <COWAN@XX.LCS.MIT.EDU>
Tue 3 Jun 86 21:07:28-EDT

The following passage from a 6-part "editorial" in the San Francisco magazine "Processed World" argues that the Space Shuttle disaster will not (as Proxmire claimed) shake people's faith in technology. Instead, it may strengthen their resolve to pursue technology regardless of risks. (Fortunately, the same argument can not be applied to the Chernobyl accident; people don't have the same love affair with Soviet nuclear power that they had with the Shuttle.)

Send me mail if you want more info about the magazine; this is from the recently published Number 16.

"Braking Star Wars, or a New Standard of Patriotism"
by Marcy Darnovsky

"If the fireball that consumed Space Shuttle Challenger slows down the development of Star Wars, the seven people that perished in it will not have died in vain.

"To millions of space enthusiasts, the shuttle and the space program are tributes to curiosity, imagination, courage, and the quest for knowledge and adventure. These are among the worthy impulses of the human spacies. But what most space boosters don't see through the glitter of the stars (leaving aside the problem of how to divide the purse between cross-town buses and interplanetary travel) is how these impulses are being used and perverted.

"Whatever its origins, there can be no doubt about what master the Shuttle now serves. Starting in 1987, the Pentagon had planned to use half of the spacecraft's cargo bay at least twice a year for Star Wars experiments alone. It had claimed a third of the available shuttle launches over the next ten years. Under the National Space Policy adopted by Reagan, the Pentagon is not only NASA's largest customer, but also its preferred customer, and as such is entitled to bump civilian, commercial, and scientific payloads off Shuttle flights.

"For a short time, the suspension of Shuttle missions and the loss of one of the four orbiters will slow the military's invasion of space. But before long, the space arms race will be back in harmony with the spheres. The scientific and commercial aspects of the space program will probably come out the losers, with NASA dancing to the Pentagon's tune even more slavishly than before.

"A month after the explosion, some of the astronauts voiced dissatisfactions with NASA safety procedures and secrecy. It's too soon to tell whether their criticisms will crack the unnerving unaniminity of popular support for more space spectaculars.

"Remarkably, instead of planting doubts about the reliability of complex technologies and the push into space, the destruction of the Challenger seems to have convinced most Americans that no sacrifice is too great for the technology that will conquer the stars. NASA reports it received 90,000 letters in the two weeks following the explosion, 99% of them supporting the space program. "Something like this brings the nation together," said Daniel Boorstin in the New York Times. "The space program in general has done that; people understand the grandeur even if not the technology, and to share that grandeur is what makes a great nation." Boorstin is right: the majestic lift-off of a rocket with human beings perched atop it raises modern Americans out of their everyday lives into an epiphany of technological awe intertwined with chauvinistic pride.

"The Shuttle catastrophe has constructed a new standard of patriotism: giving your life for your country's technology. Instead of making it acceptable to question the military takeover of space, the Shuttle disaster may make the space program more sacred than ever. If the explosion of the Challenger and the seven dead astronauts have

transformed protest into heresy, it was more of a tragedy than we've
yet realized."

Date: Wed 4 Jun 86 22:01:31-PDT
From: Peter G. Neumann <Neumann@SRI-CSL.ARPA>
Subject: Re: Unshakeable Faith in Technology: Shuttles & Nuclear Power
To: COWAN@XX.LCS.MIT.EDU
cc: RISKS@SRI-CSL.ARPA

*** Shuttle ***

Today's SF Chron contains a Los Angeles Times story by Maura Dolan:

Shuttle Program Was Doomed, Panelists Say

The space shuttle prgram was so plagued by a lack of spare parts and mission softwre and inadequate crew training that flights would have been substantially slowed or halted by now even if the Challenger disaster had not occurred, members of the presidential commission that investigated the accident said yesterday. ``There was no management of this program," a commissioner said. ``Even without the accident, the program would have ground to a halt by this point.''

The article goes on to quote other commissioners anonymously on inadequate planning, having to steal spare parts from other shuttles, lack of training time, one or two of the two simulators being down often, last-minute reprograming without testing, and so on. It also outlines some of the recommendations of the forthcoming report.

There are about four or five other ... safety things that NASA has been playing the same game with as the O-rings -- the main engine, the brakes, the flapper valves (that control fuel flow), the automatic landing system," one panelist said.

*** Nuclear Power ***

Jack Anderson's column in the same paper returned to Chernobyl and the nuclear power situation in the United States:

We have learned that, since the hideous accident in the Ukraine, the Nuclear Regulatory Commission staff called in the inspectors and informed them that new, more lenient interpretations of the fire-safety regulations had been approved by the commissioners over the inspectors' vehement protests... Incredibly, the new guidelines let nuclear plant operators sidestep the protection of redundant control systems by planing fire safety for the first set of controls only. The guidelines permit partial fire barriers between the first control system and the backup system, which can be in the same room. This means that a fire could short-circuit both systems.

Basis for SDI Assumptions?

bcsaic!douglas@uw-june <Doug Schuler>
Tue, 3 Jun 86 07:56:46 pdt

I have to question two statements that were made by Bob Estell in relation to SDI software. The first one, "A missile defense is worth having if it is good enough to save only 5% of the USA population in an all-out nuclear attack" is oft-heard. The phrase "worth having" could be applied to a number of things that aren't being had by many people (things like food, shelter, medical care, or safer cars). The question of whether something is "worth having" irrespective of costs, as if one could snap his fingers and have that thing is fine for idle conversation but of little use realistically. The question of what is worth pursuing and to what degree must be taken up by society at large. The magnitude of SDI costs as well as admitted technical dubiousness must be compared with alternatives. We can't have everything that anybody says is "worth having."

The second quote, "That shield might save 75% of the population in a terrorist attack, launched by an irresponsible source" deserves some comment. The "terrorist" argument is used fairly often also to garner support for SDI, as terrorism is a popular topic on television, etc. I am prompted to ask from what quarter this terrorist attack would arise. England? France? Also, I would expect that SDI would fail miserably in the event of anything less than the full-scale attack that it was billed as deflecting.

How does this apply to Risks? The rationale and the requirements are the basis for a system. If these are invalid, the system will probably be invalid. As Herb Lin said, "Politics are just requirements at the top level."

POSTING NUMBER 2:

[Re Bob Estell's posting]

I am not sure of the facts on this but I think it is pertinent to RISKS. What is the story on the software for the Sargent York gun? Was a "high level" language used. If so, and the complexity still defeated the project, it bodes ill for SDI which consists of [the logical equivalent of?] thousands (hundreds?) of Sargent York guns launched into space. If a high-level language was used, there is still life in the "historical" argument described by Bob Estell.

** MY VIEWS MAY NOT BE IDENTICAL TO THOSE OF THE BOEING COMPANY **

Doug Schuler (206) 865-3228
{allegra,ihnp4,decvax}uw-beaver!uw-june!bcsaic!douglas
bcsaic!douglas@uw-june.arpa

[The use of a high-level programming language is only part of the
problem. In many cases, deep flaws exist in the design, and
the implementation makes things only a little bit worse. In those
rare cases where the design is actually sound, the programming
language -- whether high-level or low-level -- introduces the
possibility of additional flaws, such as loss of encapsulation,
lack of strong typing, lack of consistent exception handling,
improper sequencing or atomic actions particularly in distributed
systems, lack of adequate control transfers and domain changes,
and so on. But such problems exist in ALL of the commonly used
programming languages. PGN]

Technical vs. Political in SDI

<LIN@XX.LCS.MIT.EDU>
Thu, 5 Jun 1986 00:32 EDT

I subscribe to RISKS, and I moderate ARMS-D. I will forward to ARMS-D
any SDI messages that appear on RISKS, unless specifically told not to
do so by the subscriber.

Peter -- Is this OK?
[SURE. FINE BY ME. Remember, I don't believe in the
alleged sharp partition between RISKS and ARMS-D. PGN]

Computer Crime Laws

Peter G. Neumann <Neumann@SRI-CSL.ARPA>
Wed 4 Jun 86 22:18:21-PDT

From the SF Chron, 4 June 1986, Washington Report, p. 13:

The house approved and sent to the Senate yesterday a bill that would
expand coverage of federal laws against computer crime.
The legislation, passed by voice vote, would make it a felony knowingly to
trespass into a "federal interest" computer -- one operated by a federal
agency, a federally insured financial institution or by stockbrokers
registered with the Securities and Exchange Commission -- to obtain anything
of value.
It also would apply to entry into private computer systems located in more
than one state. The top penalty would be five years in prison and a
$250,000 fine.
The measure also would establish a new category of misdemeanor for
"hackers" who use computer bulletin boards to display passwords to computer
systems. The top penalty would be a year in prison and a $100,000 fine.

[I note that "to obtain anything of value" does not cover denials of
service, mass deletions of data, insertion of nonbenevolent Trojan
horses, and so on. The multistate basing clause may lead some
organizations into distributed system and network operations just for
the legal coverage! PGN]

Backups for micros

<E8D%PSUVM.BITNET@WISCVM.WISC.EDU>
Wed, 4 Jun 86 09:43 EDT

There probably isn't a lot more to be said about backing-up data that is new. Since someone else brought up the subject, I'll recount a very recent case of incorrect back-up procedures from here in central PA, and then make a suggestion or two. [OK. I STILL ACCEPT A MESSAGE OR TWO ON THIS TOPIC. PGN]

A small local firm was burglarized and their micro-computers stolen. All their diskettes were also taken -- yes, including all those carefully made back-ups. I don't have exact values for the worth of the data but the loss was enough to have significant impact on a small group.

I guess this comes under the heading of improperly defining the risk. Everyone knows that computers can "eat" data and that's why one makes copies. How many of your typical users think about flood or fire, which are problems common to all data storage systems, much less theft which is a threat peculiar to micro-computer use where the diskettes are worth something -- even if they don't contain expensive programs.

I could just say, "Boy, what a dumb mistake. They should have had hard-copy of as much stuff as practical, and protected those back-up diskettes." That's not very productive, though. The answer lies in education and perhaps in program developers meeting the real needs of the users. Computer users need to know how to protect their data and why. A couple of horror-stories go a long way. Either practical back-up schemes described step-by-step (such as how to copy only files created after a certain date) or else menu type software should be generally available. This information should be easily accessible to people who don't know a whole lot about programming or even about their system. (If I were a diskette manufacturer I'd give away back-up program-packages.) And don't forget the worst part of using your archive-copies -- figuring out which version of what you are working with.

Evan Dresel
Dept. of Geochemistry   E8D @ PSUVM (bitnet)
228 Deike Bldg.   ...!psuvax1!psuvm.bitnet!e8d (uucp <-->
Penn State University   bitnet gateway)
University Park, PA 16802   e8d%psuvm.bitnet@wiscvm.arpa (arpa)
(814) 863-0672

The Clock Lies Again

Peter G. Neumann <Neumann@SRI-CSL.ARPA>
Fri 30 May 86 23:36:19-PDT

It is after midnight, but not by SRI-CSL's time. We have another clock problem. PGN [An homily anomaly?]

[This one was quite different from the one I previously reported.]

Re: The Clock Lies Again

Jagan <JAGAN@SRI-CSL.ARPA>
Sat 31 May 86 01:21:49-PDT

You are absolutely right .... However, I think the problem this time is not with the algorithm to compute the most reasonable time but the fact that the machine was unavailable (but not down!) for about half-hour this afternoon. (The clock had stopped even though the machine didn't think the clock had.) Jagan [Jagannathan]

Search RISKS using swish-e

Report problems with the web pages to the maintainer

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 2

Thursday, 5 June 1986

Contents

- Are SDI Software predictions biased by old tactical software?
  Herb Lin
- Info on RISKS (comp.risks)

Are SDI Software predictions biased by old tactical software?

<LIN@XX.LCS.MIT.EDU>
Thu, 5 Jun 1986 01:58 EDT

[Since Herb was evidently up late, since I was up late also, and since distribution of this message may stave off many overlapping responses to Bob Estell and prompt many rebuttals as well, it seems appropriate to distribute this response from Herb Lin as a special one-message issue that you can read along with RISKS-3.1. SDI is probably one of the most significant debate subjects of our lifetimes and deserves thorough coverage. Yes, it does mix politics and technology. It must. There is simply no other way. So, don't be UP IN ARMS-D. But let us keep any subsequent discussion cogent and sensible. PGN]

From:

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 3

Friday, 6 June 1986

Contents

- Watch this Space
  - Eugene Miya
- Unshakeable Faith in Technology
  - Herb Lin
- SDI as a defense against terrorists?
  - Bruce Wampler
  - Martin Moore
  - Bernie Gunther
- Basis for SDI Assumptions?
  - Herb Lin
- Info on RISKS (comp.risks)

Watch this Space

Eugene miya <eugene@ames-aurora>
5 Jun 1986 1910-PDT (Thursday)

The following is a personal observation and not an opinion of my employer.

Next week the President's Commission will be reporting its findings on the Challenger incident. Already leaks have occurred, and I find some of them in Time magazine. While I cannot completely comment on the bureaucracy problems in NASA, it is interesting to note that part of the solution to the launch decision problem is adding more members (contractors and astronauts) to the final launch decision process. There is an irony to that. One on hand we have been trying to reduce bureaucracy, to make committees smaller, and so forth, and one would ideally have astronauts and contractors "represented" by a "good" bureaucrat, and yet the solution is to increase the size and complexity of some committees. Yes, safety should be first, but how do you achieve safety? Or should I say achieve safety and balance it with complexity?

This complexity actually has another system to compare it to: SDI. I don't want to completely open a can of worms, but we should keep our eyes open on

this other space program and see how it handles complexity in contrast the
to manned space program. Several weeks ago, Danny Cohen at USC-ISI reported
somewhere (I thought it was Science, but I saw it in stronger language) that
SDI developers (i.e., the aerospace community) have been very conservative
about their use of computers and that SDI needs the state-of-the-computing-art.
Cohen said something to the effect that we have to push aerospace companies
to use the most advanced computing techniques available. Space companies
have always tried to use tried-and-true technologies and have varied them
only one slow degree at a time. I would like to point out to the
readerships of both the space and risks digests that these two different
forces are now acting upon companies like Lockheed, Rockwell, and so forth,
and it will be interesting to watch how they develop.

Both systems are quite complex, conservative to some degree, but supposedily
diverging forces are pushing for more conservativism and less conservatism.

From the Rock of Ages Home for Retired Hackers:

--eugene miya
NASA Ames Research Center
eugene@ames-aurora.ARPA
"You trust the `reply' command with all those different mailers out there?"
{hplabs,hao,dual,ihnp4,decwrl,allegra,tektronix,menlo70}!ames!aurora!eugene

---

Unshakeable Faith in Technology

<LIN@XX.LCS.MIT.EDU>
Thu, 5 Jun 1986 09:35 EDT

A small consolation is that the SDI advocates no longer use the
Shuttle as an example of the finest in American technology.

SDI as a defense against terrorists?

Bruce Wampler <unmvax!wampler@ucbvax.Berkeley.EDU>
Thu, 5 Jun 86 09:58:58 mdt

Offense is much easier than defense. The mention of terrorists
brings to mind an obvious BIG hole in the whole SDI concept. If I were a
terrorist (or even the USSR after some SDI was in place), I'd take a serious
look at the wide open U.S. society, the thousands of miles of shoreline and
the leaky borders with Mexico and Canada. Why bother trying to get through
a massive defense system (as unreliable as it might be) when you can land a
boat or drive a pickup across the border with a nuclear device and plant it
under City Hall in Anytown, USA? And if anyone has any doubts, just take a
look at the unstoppable influx of drugs and illegal aliens.

Maybe what SDI should really be is a big perimeter around our
borders to stop such things. Now if someone can just get the algorithm
to distinguish heroin, aliens, and plutonium...

Dr. Bruce E. Wampler
University of New Mexico
Department of Computer Science
Albuquerque, NM 87131

..{ucbvax | seismo!gatech | ihnp4!lanl}!unmvax!wampler

SDI as a defense against terrorists?

<mooremj@eglin-vax>
6 Jun 86 08:25:00 CDT [Hooray. A Date Appears!]

At the risk of beating a dead horse, I would like to take issue with this statement by Bob Estell:

>That shield might save 75% of the population in a terrorist attack, launched
>by an irresponsible source; this is far more likely than a saturation attack
>by a well armed power like the USSR.

The risk of such an attack (a terrorist attack with an ICBM) is nearly nonexistent. In the first place, it is a lot easier and cheaper to perform a terrorist attack, even a big one, with nothing more exotic than conventional explosives; consider, e.g., the destruction of the two main water conduits serving New York City (I just read a mediocre novel with this as its premise.)

Secondly, even if the terrorists decide to go the exotic route, chemical or biological weapons are much easier to produce (or otherwise obtain) and deliver. Several years ago someone mailed packages of white powder to various DoD sites. The powder was the crystalline form of Lance, a nerve gas; tasting the powder would cause instant death and smelling it would cause permanent brain damage.

Thirdly, even if the terrorists decide they just *have* to use an atomic bomb, it is much more practical to either build it in place (see "Build Your Own A-Bomb and Wake Up the Neighborhood" by George W. Harper in the April 1979 issue of _Analog_) or to deliver it by more conventional methods (probably ship, but possibly airplane.) It is much harder to build an effective ICBM than it is to build an effective A-bomb; a crude bomb will still do the job, but a crude ICBM will most certainly miss your target, assuming that it doesn't blow up in your face first.

Finally, even if the terrorists somehow managed to obtain a few missiles with H-bombs attached, nowhere near 25% of the US population would be endangered. At a guess, the smallest area containing 25% of the population would be the entire Boston-Washington strip, with Los Angeles, Chicago, and Atlanta (I've never liked Atlanta) thrown in for good measure. It would take a *lot* of bombs accurately delivered to kill 25% of the population. Furthermore, as Herb Lin pointed out, the technology is already there to defend against limited attacks.

Martin Moore (mooremj@eglin-vax.arpa)

SDI as a defense against terrorists?

<mck-csc!bmg@EDDIE.MIT.EDU>
Fri, 6 Jun 86 10:47:36 EDT

Libya will soon be able to buy an ICBM from Brazil. I read this in a recent article in either Time magazine or the New York Times.

How about a single missle from Cuba?

Bernie Gunther

<LIN@xx.lcs.mit.edu>
Fri, 6 Jun 1986 09:23 EDT

arms-d@xx.lcs.mit.edu
Subject: Basis for SDI Assumptions?
ReSent-To: risks@SRI-CSL.ARPA

> From: bcsaic!douglas at uw-june <Doug Schuler at uw-june> [...]
> What is the story on the software for the Sargent York gun? Was a "high level" language used? If so, and the complexity still defeated the project, it bodes ill for SDI which consists of [the logical equivalent of?] thousands (hundreds?) of Sargent York guns launched into space. If a high-level language was used, there is still life in the "historical" argument described by Bob Estell.

I don't think the Divad failed because of software, if software is construed in the narrow sense of improperly written lines of code. However, the problem WAS a system integration problem, and thus does have some relevance to software issues. The stated reason for Divad's failure was that it was unable to hit Soviet choppers at long enough range.

Consider the time that Divad shot at a latrine fan during a test, looking for the rotating blades of a helicopter. The Divad radar looked for a particular Doppler shift in the return signal, and you can imagine how the fan could mimic a helicopter blade. Is this a software problem? It seems to me that you could argue it both ways, but in either case, I don't think the presence of a high-level programming language would have helped.

[Flawed algorithms often appear as "undependable" software, although they can of course equally well be embedded in hardware. We should not try to make too much of the hardware-software distinction. The "blame" usually rests on the shortcomings of the designers and implementers... PGN]

Search RISKS using swish-e

Report problems with the web pages to the maintainer

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 4

Monday, 9 June 1986

Contents

- Re: Watch this Space
  - Mark Jackson
  - Eugene Miya
- Software developer's liability
  - Paul Schauble
- What an Algorithm!!
  - Brian Bishop
- Sgt. York's Latrine, and other stories
  - Mike McLaughlin
  - Ken Laws
- Info on RISKS (comp.risks)

Re: Watch this Space (RISKS-3.3)

<MJackson.Wbst@Xerox.COM>
9 Jun 86 10:57:10 EDT (Monday)

Your comments on the conflict between reducing bureaucracy and increasing the number of persons in the loop take us rather far afield from the risks of computer use...but they are similar to some concerns I've had for some time, and the "complexity" issue has relevance to this list, so what the heck.

In my opinion one of the *major* challenges facing humans is the need to find better ways of structuring organizations, and training individuals to function within organizations. Our present performance ranges from barely adequate to abysmal; the current consequences of this performance level are extremely serious, and the prospects are that these consequences will get worse. Blindly intoning "we need less bureaucracy" is no help.

Those are strong statements; let me explain. When the number of persons necessary to an enterprise rises much above that appropriate to a single work-group some *organizational* as opposed to *individual* division of responsibility becomes necessary. (Xerox doesn't build copiers by getting several thousand employees together, telling them all to "build copiers at a profit," and leaving them to their own devices thereafter.) As the compartmentalization of the organization increases, the relationship between the output of each unit and the goals of the organization becomes less clear. "Do your job right" becomes an unsatisfactory performance criterion; specifications become of necessity more formal. It becomes possible for individuals or sub-organizations to prosper by appearing to meet proper criteria, or by actually meeting improper criteria; such performance may actually hinder the successful fulfillment of the intended organizational goals. Individual behavior tends toward that which is *actually* rewarded by the organization, as opposed to that which is *stated* to be desired. It's like entropy; all the forces are toward declining performance, and because it's a coupled (people/structure) problem the trends are extremely difficult to reverse.

It is presently fashionable to point to the government as a bad example of rampant bureaucracy. This is to an extent fair; I believe there are two reasons that the problem is generally worse in government than in the business sector:

1) We desire of our government that it be one of "laws not of men"; this requires formal specification of acceptable performance (laws and regulations). If OSHA published simple, common-sense guidelines ("don't unduly endanger your employees") they'd be thrown out by the courts on the perfectly sound grounds that the proscribed behavior was undefined; instead we get five-page definitions of an acceptable ladder and such.

2) The constraint on organizational reasonableness which acts on business (don't be so unprofitable as to go bankrupt) is somewhat stronger than that on government (don't be so expensive and unresponsive as to cause the voters to rebel).

But the differences are those of degree, not of kind; I suspect that #1 above is the more important, and I am extremely skeptical of those who contend that a good dose of free enterprise will serve to solve, by Darwinian selection, the organizational problem. And the problem applies to not-for-profit, military, and all other "large" organizations as well.

Draw what parallels with large hardware/software systems you wish; AI buffs may note the analogy with the notorious difficulty of programming "common sense", for example.

Mark

"Absolute truth? What's that?"
"It's a five-to-four decision of the Supreme Court."
-- Dan O'Neil

Re: Watch this Space (RISKS-3.3)

Eugene miya <eugene@ames-aurora.arpa>
9 Jun 1986 1521-PDT (Monday)

I just came from a televising of Rogers and Fletcher (our own internal TV feeds). Permit me to clarify the forthcoming dilemma. The matter is not solely a problem of "bureaucracy." "Bureaucracy" is an artifact, and the word had a tainted denotation. Another, perhaps clearer artifact would be the trend in NASA from a centralized to a decentralized (NASA Centers really became "Centers") and now back to a more centralized agency (command at NASA HQ) versus the more decentralized approaches SDI (Cohen et al.) are proposing (admitted automated).

Aside: Are automated bureaucracies any better than human bureaucracies?

The gist of what I hear Mr. Jackson saying is on the nature of organizing complex systems (a la Simon's Sciences of the Artificial). I would also like to point out that Jacob Bronowski pointed out just before he died that the great challenge facing humans was the balance of individuals (I extrapolate to include centralized authority) to groups (decentralized).

The point of my posting was to note that we have an interesting juncture and we should be prepared to note the different paths taken for future comparisons (and future mis-intepresentations). Another interesting thought occurs to me about SDI, but that will be a separate note which I will Cc: to Arms-d.

Again, the viewpoints expressed are personal and not views of the Agency.

From the Rock of Ages Home for Retired Hackers:

--eugene miya
NASA Ames Research Center
eugene@ames-aurora.ARPA
"You trust the `reply' command with all those different mailers out there?"
{hplabs,hao,dual,ihnp4,decwrl,allegra,tektronix,menlo70}!ames!aurora!eugene

Software developer's liability

Paul Schauble <Schauble@MIT-MULTICS.ARPA>
Sat, 7 Jun 86 23:29 EDT

These two items are from the June 3, 1986 issue of PC WEEK.

IRS I: The Internal Revenue Service has thrown a chill over the PC software business. It recently ruled that creators of computer programs that help taxpayers prepare their tax returns may be subject to penalties if the program gives bad advice. The ruling will put the software developers on the same footing as flesh-and-blood tax advisors: at risk.

IRS II: TCS Software of Houston is already in trouble with the IRS. The company was contacted by the IRS because its tax-preparation software program, Client Tax Series-1040, was listed as the tax preparer on the 1985 tax return of one Richard P. Jamerson.

The IRS was up in arms because Mr. Jamerson had used a fictitious Social

Security number, hadn't included a check with the tax return, hadn't signed
the return or included a W-2 form. Fortunately for TCS, Mr. Jamerson owes
no taxes since he doesn't exist. He is the totally fictitious example that
goes out with the TCS package to show users how the software package works.
Apparently, one of the sample returns was inadvertently mailed to the IRS.

Paul Schauble at MIT-Multics.arpa

What an Algorithm!!

Brian Bishop <BISHOP@USC-ECL.ARPA>
Fri 6 Jun 86 14:37:26-PDT

\>-> Maybe what SDI should really be is a big perimeter around our
\>-> borders to stop such things. Now if someone can just get the algorithm
\>-> to distinguish heroin, aliens, and plutonium...

I don't know about you, but I would be much more afraid of that algorithm
than I would be of a Soviet nuclear attack.

BfB

Sgt. York's Latrine, and other stories

Mike McLaughlin <mikemcl@nrl-csr>
Fri, 6 Jun 86 16:27:59 edt

The latrine fan story keeps going around and around. The radar never saw a
latrine, much less one with a fan. The Doppler return of a hypothetical fan
on a hypothetical latrine would differ significantly from the fans on a
helicopter. The story is full of the same stuff as the latrine. Let's not
fall into it again.

[Thanks, Mike. You've got a lot of fans as we go
around in circles. "Curses, Air-foiled again?"]

Sgt York's Latrine

Ken Laws <Laws@SRI-AI.ARPA>
Mon 9 Jun 86 22:18:56-PDT

According to 60 Minutes (or was it 20/20?) the DIVAD did not shoot at a
latrine fan. It was distracted by a small ventilation fan, but I'm not sure
that it even targeted on the thing. The fan wasn't on a latrine; the
analogy to a bathroom fan was created by a PR man who was trying to explain
to reporters how small it was. The "software problem" was much easier to
fix than the PR problem.

I'm an expert-systems enthusiast precisely because such bugs do crop up in
all real-world systems. Expert systems "technology" is a form of

institutionalized hacking -- programming by successive approximation, or debugging as part of the design effort rather than part of the maintenance effort. It's related to the pancake theory ("Plan to throw the first version away. You will anyway."), but goes deeper: plan to throw every version away, but use the current one if you have to.

[Perhaps that is the radioactive pancake theory.
("They're too hot to eat, but they're fun to make.
If you really get hungry there's always one ready,
and it's probably better than starving to death.") PGN]

Effort continues on optimal algorithms and proofs of correctness, but too often we optimize the wrong thing or omit real-life complexities from our proofs. (Computers are particularly vulnerable. How do you prove that a gamma-ray burst during a critical routine won't change a crucial bit?) Those who build expert systems take the opposite tack: that systems will always contain bugs, so each piece should be robust enough to function in spite of numerous sources of uncertainty and error. This is similar to the renewed NASA policy that every critical shuttle system have a backup. I think it's a healthy viewpoint.

-- Ken Laws

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 5

Tuesday, 10 June 1986

Contents

- A powerful metal detector and magnetic personalities with bank cards
  Matthew P. Wiener
- Shuttle Launch Decisions
  Don Wegeng
- Re: Estell's defense of SDI
  Martin Purvis
- Sgt. York's Latrine, and other stories
  Mike McLaughlin
- Info on RISKS (comp.risks)

A powerful metal detector and magnetic personalities with bank cards

Matthew P. Wiener <weemba@brahms.berkeley.edu>
Sat, 7 Jun 86 02:27:54 pdt

[This item illustrates the need for awareness of the technology by people in the environment. The interference problem is also relevant to RISKS.]

In the Thursday 5 June 1986 issue of The New York Times, there is an article about an accident that occurred with a magnetic resonance imager--the first serious accident of this type.

The device uses a huge magnet with a hollow cylinder for the patient to lie inside. The accident occurred in a converted semitrailer used for mobile diagnosis. A technician was in the hollow when two steel tines, weighing more than 80 pounds each, were ripped off by the magnet from an (intentionally) approaching forklift, and ended up knocking the man 15 feet away, and breaking many bones.

The magnet complicated rescue work. A doctor could not approach until he removed his stethoscope. A paramedic's scissors flew off when he tried to cut the injured man's pants. A policeman nearly had his gun pulled from his holster. Rescuers were slow to grasp just how strong the magnetic field was, and to realize that all metal objects had to be removed in order to approach the injured. And finally--here's where the computer connection comes in!--the magnetic bank cards of the rescuers were erased.

The magnet's emergency shutdown could not be used, as it hadn't been fully installed yet. So it took 20 minutes instead.

The article pointed out that in normal usage these difficulties are not present, as normally special equipment is used and all nearby personnel are familiar with its power. But as revealed by the accident, emergency workers do not have such training. (They also do not have training for lots of special and exotic situations. There is a certain iatrogenic irony in this situation -- which is not uncommon in medical practice.)

["Iatrogenic" implies that a problem is caused or made worse inadvertently by doctors and/or medicine in what might otherwise be perceived as an attempted cure or improvement. [[As a result, one suffers from inadvertigo?]] The use of "irony" seems like an attractive pun in this context. Thanks. PGN]

Note--some details are slightly unclear from the article I read. If anyone wants more details, you are referred to a recent letter in The New England Journal of Medicine, by(?) Drs. Syverud and Fowler. -Matthew

ucbvax!brahms!weemba Matthew P Wiener/UCB Math Dept/Berkeley CA 94720

Shuttle Launch Decisions

dw <Wegeng.Henr@Xerox.COM>
10 Jun 86 09:00:41 EDT (Tuesday)

After watching the reports on TV giving the conclusions of the Rogers Commission, a question occurred to me that may be relevant to Risks. A lot of attention has been given to the fact that some of the rocket engineers recommended against launching the Challenger. What I haven't heard anyone talk about is whether such recommendations before a launch were common. The media coverage has always implied that the engineer's protests were an unusual event, but is this really the case? I can easily imagine a scenario where before every launch a different engineer recommends against launching, but management decides that their reasons are not adequate (after all, one of management's jobs is to evaluate such recommendations) and goes ahead and launches as scheduled. After awhile the situation might become similar to the little boy who cried wolf.

I'm not trying to defend NASA, or implying that the above scenario describes the situation. I'm just trying to understand the context of their decision to launch Challenger. Can anyone shed any light on this?

/Don

[I hope one of our readers can respond. With regard to the software problems, there have been complaints that the new mission software was frequently delivered only at the very last minute, and that no extensive simulation testing could be done. The impression is given that whatever the state of the software was at the final scheduled delivery date, that is what was delivered -- irrespective of how buggy it might be. I think it would be very helpful to understand the circumstances better. Tasteful reports on this subject -- as well as the more general question raised by Don -- would be welcome. PGN]

Re: Estell's defense of SDI

<CS.PURVIS@R20.UTEXAS.EDU>
Tue 10 Jun 86 21:57:50-CDT

Estell makes the following comment:

> The "complexity" and "historical" arguments even interact.
> Peter Denning observed years ago that the difficulty of understanding a program is a function of size (among other things). He speculated that difficulty is proportional to the SQUARE of the number of "units of under-standing" (about 100 lines of code). Old tactical software, in assembly language, tends to run into the hundreds of thousands of lines of code; e.g., a 500,000 line program has 5000 units of understanding, with a diffi-culty index of 25 million. That same program, written in FORTRAN, might shrink to 100,000 lines thus only 1000 units of understanding, thence a difficulty index of one million. That's worth doing!

I believe that the same program written in a "high level" language, like Fortran, would probably have about the same number "units of understanding" ~ 5000, in this case. Assuming that the "units of understanding" are understood to be higher level concepts, Fortran would enable one to write those units with fewer lines of code. But I wouldn't expect the number of those units to decline with nearly the same scale factor.

Of course the likelihood of a typographical error would be reduced by such a scale factor, but that's not the major concern here.

--Martin Purvis

Re: Sgt. York's Latrine, and other stories

Mike McLaughlin <mikemcl@nrl-csr>
Tue, 10 Jun 86 12:36:24 edt

I believe there were several retractions - enough for me to believe, at any rate. If I hadn't been so tired when I sent that bit to Peter I would have expounded further on the delightful topic of various matters hitting the fan, etc.

I *hope* that whoever designed the helicopter-rotor-selection algorithm did more than simply search for cyclic doppler. There are too many things out

in the real world that rotate but aren't helicopters.
- Wind turbines on a barn
- The rotating beacon at some airports
- Windmills
- Cooling fans on the roof of a large building
- Cooling fans on top of a diesel/electric locomotive

By the way, I have patronized a fair number of outhouses down in the Shenandoah Valley - While almost all needed (desperately!) ventilating fans, only one or two had them - and they sounded like squirrel cage blowers within a ventilating pipe, not likely to be picked up by Sgt. York's radar. Nose yes, radar no.

- Mike McLaughlin <mikemcl@nrl-csr>

[I understand that, inspired by these reports, particle physicists are now working on a new approach: Latrinos. Note: I expect that future submissions to RISKS on this subject will get flushed. (Please replace all DIVADs.) PGN]

Search RISKS using swish-e

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 6

Thursday 12 June 1986

Contents

- Risks from inappropriate scale of energy technologies
  - Michael J. Natkin
- Shuttle Software
  - David C. Smith
- An additional SDI problem: sensor technology
  - Eugene Miya
- Privacy in the electronic age
  - Dave Platt
- Sgt York software
  - Larry Campbell
  - Mark Vilain
- Info on RISKS (comp.risks)

Risks from inappropriate scale of energy technologies

"Michael J. Natkin" <mjn%brown.csnet@CSNET-RELAY.ARPA>
10 Jun 86 (Tue) 23:46:50 EDT

One of the most important categories of long term risks to the public from technology seems to have been overlooked in Risks so far. The assumption that more technology is automatically good is so ingrained in our thinking that it is hardly questioned. We measure our welfare in terms of Gross National Product, not by how many people have enough to eat, or by distribution of income.

In particular a vast amount of our technical, capital and human resources are expended developing monolithic energy technologies without regard to end use needs. The public has long been duped into the idea that centralized energy management has it's best interest in mind as we develop ever increasing electrical capacity. But centralized reactors and other "hard" technologies are extremely susceptible to terrorist attack and other failures, as has been mentioned before.

The public has been told that it doesn't have the expertise to make decisions about such high risk high technologies as SDI and nuclear power, and in some sense this is true. But the technocrats have preempted the public's right to make the moral and political policy which guides the choices.

I think that we should be pursuing a policy course which develops technology that can be put safely in the hands of non-technical people. This might take the form of small burners which use the methanol from organic wastes, windmills, or non-electrical solar collectors, to name a few possibilities. Localized, distributed technologies have many advantages, including ease of repair, localization of risk from outage, and major reductions in distribution losses and cost of distribution equipment and labor. I strongly recommend Amory Lovins' "Soft-Energy Paths" to others interested in issues of appropriate scale in technology.

Michael Natkin
CSnet: mjn@brown
ARPA: mjn%brown@csnet-relay
UUCP: ...!{allegra,decvax,ihnp4}!brunix!mjn

Shuttle Software

David C. Smith <DCSmith@SRI-AI.ARPA>
Wed 11 Jun 86 08:55:30-PDT

The cover story of the September, 1984, CACM is "A Case Study: The Space Shuttle Software System". As with other CACM case studies, this one is a discussion, or interview, with several people involved with the subject matter, in this case 6 individuals from the IBM Federal Systems Division. An Outline of the Interview included in the article contains:

Project Overview
The Shuttle Computers
Project Organization
Testing Facilities
Detailed System Operation--No Redundancy
Redundant Set Operation
System Problems
The Interprocess Variable Problem
Concluding Remarks

The issue also contains several other articles in a Special Section on Computing in Space, including "Design, Development, Integration: Space Shuttle Primary Flight Software System", written by 2 senior technicians from the IBM FSD.

It seems like a good place for a novice to the shuttle and its systems (like myself) to get some basic information about the shuttle computers and the complexity of the systems.

Dave Smith

An additional SDI problem: sensor technology

Eugene Miya <eugene@ames-aurora.arpa>
11 Jun 1986 1124-PDT (Wednesday)

The view expressed within are the view of the author and not of my agency nor of the Federal government. ----------------------------- A lot of interest has been expressed regarding the focus of the problems of SDI: the software, in particular battle management. Note the Science article of May 9 1986. However, I wonder about the other components of the system. Where there are various groups watchdogging computing, but the more hardware oriented, EE areas such as radar have fewer opposition elements. Recent postings on cruise missiles and the integration of DIVAD move me to post this.

Sensor technology is one area which worries me. SDI battle management makes certain assumptions about the ability to detect and identify targets. I think that most computer people don't understand the nature of radar to worry about the problems of `target' detection and ranging. That is all that radar is: detection (boolean) and ranging (distance=rate times time). A first starting references is Skolnick's text on Radar. (Dated)

Inherent problems with a ranging system include: Range and azimuth ambiguities, difficulties with empirically determined signatures. Most people don't seem to understand that knowing the geometry of systems are important. Satellite images [some radar maps to be used in offensive missiles] are not photographs (you must call them images) because their geometry is from a linear and not a point perspective, so distance determination for things like cruise missiles cannot be done using a straight edge. Radar (simple) is like looking at the world using a monochromatic spot light from the point where you are looking: you don't get shadows (an important distance cue). Note: I have not talked about clutter, or noise (ever wonder how high speed jets detect jets from ground objects, or how AWACS which points down get insignificant ground objects cleared?).

While there exist solutions, all of them involve tradeoffs in complexity, cost, and new emergent problems. Solutions in Doppler systems, phased arrays, stereo transmit/receive systems, but just the inherent simplicity of the concept and the over-generalization of use worries me. This is a case where "high-level language" solutions may not be high-enough.

--eugene miya, NASA Ames Research Center, eugene@ames-aurora.ARPA
{hplabs,hao,dual,ihnp4,decwrl,allegra,tektronix,menlo70}!ames!aurora!eugene

Privacy in the electronic age

Dave Platt <Dave-Platt%LADC@HI-MULTICS.ARPA>
Wed, 11 Jun 86 10:47 PDT

A news clipping from this morning's "Los Angeles Times" (page 2, The News in Brief):

The House Judiciary Committee voted 34 to 0 for a bill seeking to
bring constitutional guarantees of the right to privacy into the
electronic age. The legislation would extend laws that now protect
the privacy of the mails and land-line telephone conversations to also
cover electronic mail and some telephones that use radio waves.
The bill was cleared at the request of Rep. Robert W. Kastenmeier
(D-Wis.), chairman of Judiciary's subcommittee on courts, civil
liberties and administration of justice.

Anyone know the details? Just what privacy coverage would be afforded
by this bill in its present form? How would the bill's provisions
affect the sysops of private electronic bulletin-board systems, for
example? Would this bill clarify the legal standing of electronic
transactions and messages re their use as evidence in court?

[Very strange. RISKS-3.1 noted that the House sent a bill to the
Senate on 3 June that covered "federal interest" computers. Is this
an additional bill, or a modification of one already sent over?
Maybe someone in the House is reading RISKS and noted the apparent
flaws in the bill that I mentioned in RISKS-3.1? PGN]

Sgt York software

<decvax!bellcore!genrad!panda!wjh12!maynard!campbell@ucbvax.berkeley.edu>
Wed, 11 Jun 86 01:52:39 edt

In RISKS 3.4, Mike McLaughlin (mikemcl@nrl-csr) and Ken Laws (laws@sri-ai)
dispute the Sargent York latrine fan story. [...]

I quote from a story by Gregg Easterbrook in the November 1984 issue of
The Washington Monthly:

> During a test one DIVAD locked on to a latrine fan. Michael Duffy,
> a report for the industry publication _Defense Week_, who broke this
> aspect of the story, received a conference call in which Ford officials
> asked him to describe the target as a "building fan" or "exhaust fan"
> instead.

The Washington Monthly and _Defense Week_ are both reputable publications.
Does anyone have a citation for a retraction in _Defense Week_, or should we
assume that the TV networks swallowed Ford's story whole?

Larry Campbell The Boston Software Works, Inc.
ARPA: campbell%maynard.uucp@harvard.ARPA 120 Fulton Street, Boston MA 02109
UUCP: {alliant,wjh12}!maynard!campbell (617) 367-6846

Sgt. York software

Marc Vilain <MVILAIN@G.BBN.COM>

Wed 11 Jun 86 12:48:29-EDT

Here is some information on the DIVAD software that hasn't appeared yet in this forum. [It] is abstracted from a longer note compiled by Reid Simmons from material he received from Gregg Easterbrook (both his article in the Atlantic, and personal communications).

According to Easterbrook, the DIVAD did target a latrine exhaust fan in one series of tests. The target was displayed to the gunners that man the DIVAD. But the Sgt. York did not shoot at the latrine, or even swivel its turret in the latrine's direction, having prioritized the target as less important than other targets in its range.

In another series of tests (Feb. 4 1984), U.S. and British officials were to review the DIVAD as it took upon a rather cooperative target: a stationary drone helicopter. On the first test run, the DIVAD swiveled its turret towards the reviewing stand as "brass flashed" and the officials ducked for cover. It was stopped only because an interlock was put in place the night before to prevent the turret from being able to point at the reviewing grandstand. Afterwards, the DIVAD shot in the general direction of the helicopter but the shells traveled only 300 yards. The official explanation is that the DIVAD had been washed the night before, screwing up its electronics. Easterbrook wonders what would happen if it rained in Europe when the DIVAD was being used.

Easterbrook goes on to claim that the snafus the DIVAD experienced were very much due to software. The main problem was that the pulse-Doppler tracking radar and target acquisition computer were a very poor match. Easterbrook claims that the hard problem for the software (tracking fast, maneuvering planes) was easiest for the pulse-Doppler radar which needs a moving target. On the other hand, the hard part for the radar (detecting stationary helicopters) was the easiest to aim at. The DIVAD mixed two opposing missions.

Easterbrook goes on to say that human gunners are often more successful than their automated counterparts. They can pick up on visual cues, such as flap position on approaching aircraft, to determine what evasive maneuvers the enemy might make. These kinds of cues are not visible to things like pulse-Doppler radars. Further, evasive courses of action are hard for human gunners to counter, but even harder for target tracking algorithms (again the lack of visual cues comes as a disadvantage). For example, the DIVAD expected its targets to fly in a straight line (which my military friends tell me is not too likely in a real combat).

There is lots more to the Sgt. York story, not all of which is relevant here. If there is a moral to be drawn specifically for RISKS, it's that as advanced as our technology may be, it may not always be the match of the problems to which it is applied. This was certainly the case with the unfortunate DIVAD.

marc vilain

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Index to Volume 2

Saturday, 31 May 1986

Issue 1 (1 Feb 86)

- First Six Months of the Forum in Retrospect; *** Updated Disaster List *** (Peter G. Neumann)

Issue 2 (1 Feb 86)

- More on Shuttle destruct systems (Martin J. Moore, Sean Malloy, Brint Cooper)
- The Challenger [non]accident (Herb Lin)
- Redundancy (D. Cook)
- Galileo Plutonium power (Martin Schoffstall, James Tomayko)
- VDT's and birth defects in mice (Dan Hoey)
- ORCON dissemination constraint on RISKS 1.43 (Ted Lee)

Issue 3 (1 Feb 86)

- The possible vs the impossible (Dave Parnas)
- RISKS generalizations (Jim Horning)
- Challenger speculation (Henry Spencer)
- Possible triggering of the self-destruct mechanism (Don Wegeng)
- Redundancy in the Shuttle's Computers (Mark S. Day)
- Galileo Plutonium power (Herb Lin)
- Icing the Shuttle (Jim McGrath)

Issue 4 (2 Feb 86)

- Solid propellants (Mike McLaughlin)
- Plutonium (Jim McGrath)
- SRB Self-Destruct Mechanisms (Clive Dawson)
- Details on the 1981 Quebec election -- a program bug (Jean-Francois Lamy)

Issue 5 (3 Feb 86)

- SRBs and What the Computers Should Monitor (Sean Malloy, Charley Wingate)
- SRB survival (Bill Keefe)
- Physical Security at the Cape (Tim Wicinski)
- A hard rain is gonna fall, (Marc Vilain)
- Correction re Galileo plutonium (James Tomayko)
- Quebec Election (Dan Craigen)
- SCRIBE time-bomb goes off! (Peter G. Neumann)

- Issue 6 (4 Feb 86)
  - Shuttle computers (Marc Vilain) -- from NY Times
  - SRBs and Challenger (Mike Iglesias) -- from LA Times
  - Galileo, Plutonium, Centaur, physical security [4 messages] (Henry Spencer)
  - RISKS-2.5 & "Some simple calculations" (Bob Ayers)
  - A hard rain is gonna fall. (Herb Lin)
  - By the slip of a finger ... (Ted Lee)
- Issue 7 (6 Feb 86)
  - The lesson of Challenger (Barry Shein)
  - Mistaken Arrest due to computer error (Steve Rabin)
  - Denial of [Religious] Service (Chris Guthrie)
  - Earthquake Monitoring Systems (Gary T. Leavens)
  - Mice & CRT Radiation (Ted Shapin)
  - SRBs, What the Computers Should Monitor, and Expert Systems? (Jim Giles)
  - Redundancy in the Shuttle's Computers (K. Richard Magill)
  - Nuclear Cargo in the Shuttle (Larry Shilkoff)
  - Software Protection Symposium (Barbara Zayas)
- Issue 8 (7 Feb 86)
  - Expert systems and shuttles (Michael Brown, Dave Platt)
  - Plutonium (Martin J. Moore)
  - Earthquake Monitoring Systems (Mike Raugh via Matt Bishop, Hal Murray, Eugene Miya)
- Issue 9 (9 Feb 86)
  - Computerized train wreck? ... Computer-induced stock-market swings. (Martin Minow)
  - Selectively Displaying Data -- Boeing 767 EFIS (Alan M. Marcum)
  - Cape Range Safety Display Systems (Lynne C Moore)
- Issue 10 (12 Feb 86)
  - Computerized aircraft collision avoidance (Peter G. Neumann)
  - Computerized Feedback and the Stock Market (Eric Nickell)
  - Analyst Changes City Treasurer's Computer Code (Mike McLaughlin)
  - Plutonium on the Space Shuttle (Tom Slone)
  - Request to RISKS Readers from COMPASS 86 (COMPuter ASSurance) (Al Friend)
- Issue 11 (16 Feb 86)
  - SF Federal Reserve Bank 2 Billion Dollar Goof (SF Chron via Peter G. Neumann)
  - Washington D.C. Analyst's Password Game (AP via Geoff Goodfellow)
  - Boeing 767 EFIS -- compare Airbus A320 (Rob Warnock)
  - Networks Pose New Threats to Data Security (InfoWorld-86/2/10 via Werner Uhrig)
- Issue 12 (18 Feb 86)
  - Risks in automobile microprocessors -- Mercedes 500SE (Peter G. Neumann)
  - Train safeguards defeated (Chuck Weinstock)
  - Security Safeguards for Air Force Computer Systems (Dave Platt)
  - How can Alvin Frost fight City Hall? (Jim DeLaHunt)
  - More Plutonium/Shuttle (Martin J. Moore)
  - Computerized Voting -- talk by Eva Waskell (Wednesday eve, 19 February, MIT)

- Issue 13 (20 Feb 86)
  - Dec. 8 cruise missile failure caused by procedural problems (Martin J. Moore)
  - Computerized voting (Matt Bishop)
  - Non-science quotations on Plutonium (Bob Ayers)
  - Software Piracy (D.Reuben)
  - Air Force Security Safeguards (Stephen Wolff)
  - Shuttle Safety (NYTimes News Summary)
- Issue 14 (24 Feb 86)
  - Automotive Problems Intensify (Peter G. Neumann)
  - A hard rain is gonna fall (around March 23) (Martin J. Moore)
  - Misdirected modems (Alan Silverstein)
  - Witch hunts, or Where does the buck stop? (M.L. Brown)
  - Spells and Spirits (Steve Berlin)
- Issue 15 (25 Feb 86)
  - Software Safety Survey (Nancy Leveson)
  - Titanic Effect (Nancy Leveson)
  - F-18 spin accident (Henry Spencer)
  - Space shuttle problems (Brad Davis)
  - Misdirected modems (Matt Bishop)
- Issue 16 (25 Feb 86)
  - Volunteers to study security of computerized voting booths? (Kurt Hyde)
  - Our Economy Is Based On Electricity (Jared M. Spool)
  - Misdirected modems (Jared M. Spool)
  - The Titanic Effect (Earl Boebert)
- Issue 17 (28 Feb 86)
  - Replacing humans with computers? (Nancy Leveson)
  - Eastern Airlines stock (Steve Strassmann)
  - Computerized stock trading and feedback systems (Kremen)
  - Computer Voting Booths (Larry Polnicky)
  - Reliance on security (Jong)
  - AI risks (Nicholas Spies)
  - Data Encryption Standard (Dave Platt)
- Issue 18 (28 Feb 86)
  - Titanic and What did I overlook? (Hal Murray)
  - Titanic Effect (Jong)
  - Computers placing telephone calls (Art Evans)
  - Misdirected modems (Sam Kendall)
  - Modems and phone numbers (David Barto)
  - Misdirecting my modem (Mike McLaughlin)
  - Power-outages, & other failures of central DP systems (Dave Platt)
  - Computer voting booths (Dave Platt)
  - Data Encryption Standard (Chris McDonald)
- Issue 19 (2 Mar 86)
  - A word from Isaac Asimov about Robots (Bryan)

- AI risks (John Shore)
- Replacing Humans with Computers (David desJardins)
- On-line Slot Machines (Jeff Makey)

Issue 20 (2 Mar 86)

- Risks in Encryption (Jerry Saltzer)
- NSA and encryption algorithms (Curtis Jackson)
- Low-Tech Computerized Voting (Harry S. Delugach)
- Risks in ballot-counting systems (Larry Campbell)
- Misdirected modems (Richard H. Lathrop)

Issue 21 (3 Mar 86)

- The risks of (not) using Robots (Hal Murray)
- Computerized Voting Booths (Larry Polnicky)
- No-carrier detection by misdirected modems (Dave Platt)

Issue 22 (5 Mar 86)

- Voting receipt (Mike McLaughlin)
- Voting booths (Jim McGrath)
- Computerized Voting (Tom Benson)
- Replacing humans with computers (Alan M. Marcum)
- Electricity's power (Marianne Mueller)

Issue 23 (6 Mar 86)

- Computerized voting (Jeff Mogul, Larry Polnicky, Peter G. Neumann)
- ATM Ripoff (Dave Curry)
- Internet importance/robustness (Tom Perrine)

Issue 24 (8 Mar 86)

- Computerized ballot stuffing (Andy Kegel)
- Progress report on computerized voting (Kurt Hyde)
- Wild Modems (Bjorn Benson)
- Misdirected modems (Phil Ngai)
- Power outages (Phil Ngai)
- Earthquake problems with Nuclear Reactors (Lindsay F. Marshall)

Issue 25 (10 Mar 86)

- Balloting (Barbara E. Rice)
- Canceling ballots (Jim McGrath)
- Bank robbery (Curtis Jackson)
- Earthquake problems with Nuclear Reactors (throopw)
- Modems DON'T WORK AS SUPPOSED (Brent Chapman, Martin J. Moore, Phil Ngai)

Issue 26 (14 Mar 86)

- Integrity of the Electoral Process (Mark Jackson)
- Ballot Secrecy (Lindsay F. Marshall)
- Nuclear waste-land (Jerry Mungle)
- Nuclear disasters (Lindsay F. Marshall)
- 103/212 modems (Ephraim)

Issue 27 (15 Mar 86)

- Overload of a different sort [Air traffic stoppage] (Ted Lee)
- Cordless Phones Cry Wolf! (Peter G. Neumann)
- The Mob Breaks into the Information Age (Mike McLaughlin)
- [Non]computerized train wreck (Mark Brader)
- Ballot Integrity; Specialization in Decision-Making (Tom Benson)
- Network Security, Integrity, and "Importance" (Kurt F. Sauer)
- Modems (James R. McGowan)

Issue 28 (17 Mar 86)

- Risks of commission vs. risks of omission (Dave Parnas and Peter G. Neumann)
- The TIME is RIPE -- a clock problem (Peter Neumann)
- Mailer Gone Mad? (Landrum)
- Money Talks (Matthew Kruk)
- Another discourteous modem (Glenn Hyatt)
- Will the modem discussions ever hang up? (Rob Austein)

Issue 29 (17 Mar 86)

- Commission vs. Omission (Martin J. Moore plus an example from Dave Parnas)
- A Stitch in Time (Jagan Jagannathan)
- Clockenspiel (Jim Horning)
- Cordless phones (Chris Koenigsberg)
- Money talks (Dirk Grunwald, date correction from Matthew Kruk)
- [Non]computerized train wreck (Mark Brader)
- On-line Safety Database (Ken Dymond)

Issue 30 (18 Mar 86)

- Classes of Errors (Scott Rose)
- Range Safety System (David desJardins)
- Commission vs omission (Geoffrey A. Landis)
- Stupid Clock Software (Dave Curry)
- Control characters in headers from eglin-vax (Martin J. Moore)
- Money Talks (Prasanna G. Mulgaonkar)

Issue 31 (19 Mar 86)

- Still more on shuttle destruct systems (Martin J. Moore)
- Clock Synchronization (Andy Mondore)
- Timestamp integrity at system startup (John Coughlin)
- Danny Cohen on SDI (Charlie Crummer)
- Two more mailer problems (Sidney Markowitz)
- Marking money for the blind (Atrocity Joelll)
- Why would anyone want to computerize voting? (Larry Campbell)

Issue 32 (20 Mar 86)

- Om/Comm-ission, and analysis of risks (Niall Mansfield)
- RSO's and IIP's (Dave Curry)
- Complex systems ru(i|n)ning our cities (Mike Mc Namara)
- Re: Two more mailer problems (Bernard S. Greenberg)
- Banknotes for the visually handicapped (Nigel Roberts, Barbara E. Rice)
- Psychological and sociological consequences (Harald Baerenreiter)

Issue 33 (23 Mar 86)

- RSO's and IIP's - Martin Moore's response (Dave Curry)
- Omissions/commissions and missile destructs (Chris McDonald)
- Blind and Paper Money (sdo)
- Two Cases of Computer Burglary (NY Times)

Issue 34 (27 Mar 86)

- RSO's and IIP's - Martin Moore's response (Henry Spencer)
- Range Safety: a final word (Martin Moore)
- Someone really sophisticated, with a Ph.D... (Nigel Roberts, Keith F. Lynch)

Issue 35 (30 Mar 86)

- San Jose Library (Matthew P. Wiener, Ken Laws)
- Inter-system crashes (Rich A. Hammond)

Issue 36 (1 Apr 86)

- Errant Clocks (Barry Shein)
- Computer Illiteracy (Matthew P. Wiener)
- San Jose Library (Dick Karpinski, Holleran)
- Psychological and sociological consequences (Dave Benson)
- More inter-system crashes (Henry Spencer)
- COMPASS 86: A Progress Report (Al Friend)

Issue 37 (6 Apr 86)

- Request for information about military battle software (Dave Benson)
- Programming productivity (Henry Spencer)
- Space Shuttle Software (via PGN)
- Open-and-Shut Case Against Reagan's Command Plane (Geoffrey S. Goodfellow)
- Computer Illiteracy (Matt Bishop)

Issue 38 (8 Apr 86)

- The UK Driving Vehicle Licensing Centre (Brian Randell)
- Computer crime wave (Chris Hibbert)
- Programming productivity (Herb Lin)
- Request for information about military battle software (Scott E. Preece)
- Aviation Week Technical Survey: AI & Aviation (Werner Uhrig)

Issue 39 (11 Apr 86)

- $36 million accounting mistake (Graeme Hirst)
- Admissability of computer files as evidence? (Kathryn Smith)
- "Rapid advance" of SDI software (Walt Thode)
- Blame-the-computer syndrome (JAN Lee)
- Hackensack Phone Snafu (Dirk Grunwald)

Issue 40 (12 Apr 86)

- GREAT BREAKTHROUGHS [Red Herrings swimming upstream?] (Dave Parnas)
- Military battle software ["first use", "works"] (James M Galvin, Herb Lin, Scott E. Preece, Dave Benson)
- First use - Enterprise (Lindsay F. Marshall)

Issue 41 (13 Apr 86)

Computer Naivete (Lindsay F. Marshall)
- Admissability of computer files as evidence (Scott E. Preece)
- Programming productivity (Henry Spencer)
- The San Jose Public Library [and responsibilities] (Sriram Vajapeyam)

Issue 42 (14 Apr 86)

- Robot safety (Ron Cain via Bill Park)
- Use of computer files as evidence (Rob Horn)
- Review of *Softwar* (Gary Chapman)
- Computerized Voting -- No Standards and a Lot of Questions (Summary of Eva Waskell's talk by Ron Newman)

Issue 43 (17 Apr 86)

- Re: Review of *Softwar* (Marvin Schaefer)
- GREAT BREAKTHROUGHS (Herb Lin)
- Star Wars software advance (AP)
- Smart bombs in Libya (Washington Post)
- Pacific Bell Bills (SF Chronicle)
- BU joins the InterNet... (Barry Shein)

Issue 44 (21 Apr 86)

- Why Simulation Is A Good Thing... (Lynne C. Moore)
- Hacking & forgery laws (Robert Stroud)
- Strategic Systems Reliability Testing (Dan Ball)
- SDI (Larry Campbell)
- Cost of phone billing error (Dave Redell)
- Normal Accidents and battle software (Dave Benson)
- Psychological risks, part II (Dave Benson)

Issue 45 (28 Apr 86)

- HBO gets Hacked:: We Interrupt This Program ... for a Viewer Protest. (Geoff Goodfellow, Frank J. Wancho)
- Ball's contribution on Polaris and SDI (from Dave Parnas)
- SDI Reliability Testing - Offensive deterrent vs SDI (Jon Jacky)
- What are the limits to simulation? (Eugene Miya)
- Reference on admissibility of computer records (Bill Cox)
- Phone billing error at Pacific Bell, etc. (John Coughlin)
- Cracked Libya Defense (Udo Voges)
- Challenger article (Ron Minnich)

Issue 46 (29 Apr 86)

- Martin J. Moore (on Challenger article)
- TV "piracy" (Nicholas Spies)
- HBO -- Hacked Briefly Overnight (Mike McLaughlin)
- The dangers of assuming too much -- on TMI-2 (J. Paul Holbrook)
- A POST Script on Nuclear Power (Peter G. Neumann)

Issue 47 (1 May 86)

- HBO hacking (Phil R. Karn, Dan Franklin)
- What are the limits to simulation? (Herb Lin)
- Strategic Systems Reliability Testing (Herb Lin)
- Correction on Challenge Discussion (Jeff Siegal)

- Issue 48 (3 May 86)
  - Failure to Backup Data (James H. Coombs)
  - Computer detracting from effective communication? (Bruce A. Sesnovich)
  - Words, words, words... (Mike McLaughlin)
  - Copyright Laws (Matthew Kruk)
  - Re: Correction on Challenger (Martin J. Moore)
- Issue 49 (7 May 86)
  - Perrow on reactor containment vessels (Richard Guy)
  - Captain Midnight (Scott Dorsey, MRB)
  - NSA planning new data encryption scheme - they'll keep the keys (Jon Jacky)
  - Espionage (Mike McLaughlin)
  - The Star Wars Swindle (Dave Weiss)
  - Backups (Will Martin)
  - Interpreting Satellite Pictures (Lindsay F. Marshall)
  - Word-processing damages expression (Niall Mansfield, PGN)
  - Proofreading vs. computer-based spelling checks (Dave Platt)
- Issue 50 (8 May 86)
  - Refocus the discussion, please! (Bob Estell)
  - [Response.] Also, Delta rocket shutdown (Peter G. Neumann)
  - Large systems failures & Computer assisted writing (Ady Wiernik)
  - DESisting (dm, William Brown II)
  - Failure to Backup Data (Greg Brewster)
- Issue 51 (11 May 86)
  - Reliability limits (Brian Randell)
  - NSA assigning encryption keys (Jay Elinsky)
  - HBO pirate (Lauren Weinstein)
  - Failure to Backup Data, by James H. Coombs (Roy Smith)
  - Admissibility of legal evidence from computers (Mike McLaughlin)
  - Electronic document media (Mike McLaughlin)
- Issue 52 (13 May 86)
  - Launch failures (Phil R. Karn)
  - Brittleness of large systems (Dave Benson)
  - HBO (Scott Dorsey, Dave Sherman)
  - Word processing -- reroute [reroot?] the discussion (Chuq Von Rospach)
- Issue 53 (16 May 86)
  - A late report on the Sheffield (AP [from Martin Minow], LA Times [Dave Platt]
  - News items [Lobsters; Eavesdropping] (Alan Wexelblat)
  - More Phone Bill Bugs... (Dave Curry)
  - Backup problems (Davidsen, Roy Smith)
- Issue 54 (25 May 86)
  - Meteorites (Larry West)
  - Meteorites, Chernobyl, Technology, and RISKS (Peter G. Neumann)
  - London Stock Exchange Computer System Crash (Lindsay F. Marshall)
  - Backup (Fred Hapgood, Bruce O'Neel)

Issue 55 (28 May 86)

- Culling through RISKS headers; SDI (Jim Horning)
- Blind Faith in Technology, and Caspar Weinberger (Herb Lin)
- Risks of doing software quality assurance too diligently (PGN from Chris Shaw and the Torrance Daily Breeze)
- Collegiate jungle (Mike McLaughlin)
- Decease and Desist -- Death by Computer (Deborah L. Estrin)
- The Death of the Gossamer Time Traveler (Peter G. Neumann)
- Computer Ethics (Bruce A. Sesnovich)

Issue 56 (30 May 86)

- A joke that went wrong (Brian Randell)
- Computer Program for nuclear reactor accidents (Gary Chapman)
- On risks and knowledge (Alan Wexelblat) [Excerpt]
- Technical vs. Political in SDI (Dave Benson)
- Are SDI Software predictions biased by old tactical software? (Bob Estell)
- Culling through RISKS headers (Jim Horning)

THE RISKS DIGEST

Forum On Risks To The Public In Computers And Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Search RISKS using swish-e

The RISKS Forum is a moderated digest. Its USENET equivalent is comp.risks. (Google archive)

- Vol 26 Issue 47 (Monday 6 June 2011) <= Latest Issue
- Vol 26 Issue 46 (Saturday 4 June 2011)
- Vol 26 Issue 45 (Tuesday 24 May 2011)

- News about the RISKS web pages
- Subscriptions, contributions and archives

Feeds

RSS 1.0 (full text)

RSS 2.0 (full text)

ATOM (full text)

RDF feed

WAP (latest issue)

Simplified (latest issue)

Smartphone (latest issue)
Under Development!!

You can also monitor RISKS at Freshnews, Daily Rotation and probably other places too.

Please report any website or feed problems you find to the website maintainer. Report issues with the digest content to the moderator.

Selectors for locating a particular issue from a volume

Volume number: Volume Issue Number: Issue

Get Specific Issue

Volume Index

The dates and counts do not include the index issues for each volume.

Index to the RISKS Digest

Volume Number	Date Range	Number of Issues
Volume 1	*1 Aug 1985 - 31 Jan 1986*	*45 issues*
Volume 2	*1 Feb 1986 - 30 May 1986*	*56 issues*
Volume 3	*4 Jun 1986 - 30 Oct 1986*	*91 issues*
Volume 4	*2 Nov 1986 - 6 Jun 1987*	*96 issues*
Volume 5	*7 Jun 1987 - 31 Dec 1987*	*84 issues*

Volume 6	2 Jan 1988 - 31 May 1988	94 issues
Volume 7	1 Jun 1988 - 22 Dec 1988	98 issues
Volume 8	4 Jan 1989 - 29 Jun 1989	87 issues
Volume 9	6 Jul 1989 - 30 May 1990	97 issues
Volume 10	1 Jun 1990 - 31 Jan 1991	85 issues
Volume 11	4 Feb 1991 - 28 Jun 1991	95 issues
Volume 12	1 Jul 1991 - 24 Dec 1991	71 issues
Volume 13	6 Jan 1992 - 2 Nov 1992	89 issues
Volume 14	4 Nov 1992 - 27 Aug 1993	89 issues
Volume 15	2 Sep 1993 - 29 Apr 1994	81 issues
Volume 16	2 May 1994 - 22 Mar 1995	96 issues
Volume 17	27 Mar 1995 - 1 Apr 1996	96 issues
Volume 18	5 Apr 1996 - 31 Mar 1997	96 issues
Volume 19	1 Apr 1997 - 23 Sep 1998	97 issues
Volume 20	1 Oct 1998 - 31 Jul 2000	98 issues
Volume 21	15 Aug 2000 - 29 Mar 2002	98 issues
Volume 22	1 Apr 2002 - 27 Oct 2003	98 issues
Volume 23	7 Nov 2003 - 2 Aug 2005	96 issues
Volume 24	10 Aug 2005 - 30 Dec 2007	93 issues
Volume 25	7 Jan 2008 - 1 Apr 2010	98 issues
Volume 26	8 Apr 2010 - 6 Jun 2011	47 issues

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Index to Volume 4

Saturday, 6 June 1987

Volume 4 Issue 1 (2 Nov 86)

- Latest version of the computer-related trouble list (Peter G. Neumann)

Volume 4 Issue 2 (2 Nov 86)

- Insurgent Squirrel Joins No-Ways Arc (Ross McKenrick)
- Collision avoidance systems - FAA vs. Honeywell (Charlie Hurd)
- The Military and Automatic Humans (Ronald J Wanttaja)
- Assessing system effectiveness (Scott E. Preece)
- Computers in elections (Kurt Hyde)
- 17th FAULT-TOLERANT COMPUTING SYMPOSIUM (Flaviu Cristian)

Volume 4 Issue 3 (3 Nov 86)

- The Big Bang at the London Stock Exchange (Jonathan Bowen)
- UK computer security audit (Robert Stroud)
- Austin's computer-controlled traffic lights (Alan Wexelblat)
- Computers and Medical Charts (Elliott S. Frank)

Volume 4 Issue 4 (4 Nov 86)

- Flawed Radars in Air Traffic Control (PGN/UPI)
- The Future of English (risks of technocrats, risks of word processors) (Martin Minow)

Volume 4 Issue 5 (5 Nov 86)

- Computer causes chaos in Brazilian Election (Jonathan Bowen)
- Risks of FAA Philosophy ? (Robert DiCamillo)
- Computers and Medical Charts (Christopher C. Stacy)
- Re: Insurgent Squirrel Joins No-Ways Arc (rsk)
- Micros in Car engines (Peter Stokes)

Volume 4 Issue 6 (6 Nov 86)

- Computerized Reagan swamps Hospital with calls (David Whiteman via Werner Uhrig)
- Aftermath of the Big Bang (Robert Stroud)
- Fault tolerant computer manufacturer RISKS (Robert Stroud)
- Re: Micros in Car engines (Don Wegeng)

- Re:airplanes and risks, Risks 3.89 (Udo Voges)

Volume 4 Issue 7 (7 Nov 86)

- Risks of RISKS (PGN)
- Details on the British Air Traffic Control computer outage (from Herb Hecht)
- Re: UK computer security audit (Robert Stroud)
- USS Liberty (Matthew P Wiener)
- Grassroots sneak attack on NSA (Matthew P Wiener)
- A variation of the Stanford breakin method (Arno Diehl)
- Re: Subject: Computers and Medical Charts (Roy Smith)
- DDN Net breakdown (?) on 6 Nov 86? (Will Martin)
- Re: Linguistic decay (Matthew P Wiener)
- Mechanical Aids to Writing (Earl Boebert)

Volume 4 Issue 8 (9 Nov 86)

- Brazilian laws require proof of voting. People NEED those cards. (Scot E. Wilcoxon)
- Grassroots sneak attack on NSA (Herb Lin, Matthew P Wiener)
- Ethernet Security Risks (Phil Ngai)
- Perfection (Herb Lin)
- Information replacing knowledge (Daniel G. Rabe)
- Word Processors / The Future of English (Stephen Page)
- Copyrights; passwords; medical information (Matthew P Wiener)

Volume 4 Issue 9 (10 Nov 86)

- Risk of aging (Lee F. Breisacher)
- Re: UK computer security audit (Henry Spencer)
- Lost files (Norman Yusol)
- Canard!! [Looping Mailers] (Lindsay F. Marshall)
- Friend-foe identification (Henry Spencer)
- Micros in Car Engines (Jed Sutherland)
- Information replacing knowledge (Bard Bloom, Herb Lin, Jerry Saltzer)
- Spelling becoming obsolete? (Ted Lee)
- They almost got me! [A motor-vehicle database saga] (Mark Hittinger)

Volume 4 Issue 10 (12 Nov 86)

- Extreme computer risks in British business (Lindsay F. Marshall)
- Alabama election snafu caused by programmer (PGN)
- Looping mailer strikes again (Brian Reid, Nancy Leveson)
- Lost files on Bitnet (Niall Mansfield)
- VOA car testing (Bill Janssen)
- Re: Aftermath of the Big Bang (apology) (Robert Stroud)
- Re: The Future of English (T. H. Crowley [both of them])
- Word-processors Not a Risk (Ralph Johnson)

Volume 4 Issue 11 (14 Nov 86)

- Computers don't kill people, people kill people (Howard Israel)
- Open microphone in the sky (Bob Parnass)
- Computerized Voting in Texas (Jerry Leichter)
- Problems with HNN (Alan Wexelblat)
- Post-hacker-era computer crime (Talk by Sandy Sherizen)
- Re: They almost got me! [A motor-vehicle database saga] (Doug Hardie)

- Re: information replacing knowledge (G.L. Sicherman)

- Volume 4 Issue 12 (16 Nov 86)
  - Air Traffic Control radar problems
  - Stuck Microphone and Near-Collision of 727s
  - Gwinnett County Voting (Scott Dorsey)
  - Micros in cars (Paul Kalapathy)
  - DMV computer networks (Bob Campbell)
  - Serious security bug in 3.4 (Dave Martindale)
  - "Maj. Doug Hardie" and his story (Bruce Schuck)
  - Necessity of language skills (Daniel G. Rabe)
  - Call for Papers -- Safety and Reliability Society Symposium (Nancy Leveson)

- Volume 4 Issue 13 (18 Nov 86)
  - Framing of life-and-death situations (Jim Horning)
  - On placing the blame (Peter J. Denning)
  - Computer picks wife (Matthew Kruk)
  - Re: Micros in cars (Brint Cooper)
  - Re: They almost got me! (Will Martin)
  - Re: A variation of the Stanford breakin method (Joe Pistritto)
  - Microfiched income-tax records stolen (John Coughlin)
  - Re: Copyrights (Andrew Klossner)

- Volume 4 Issue 14 (19 Nov 86)
  - Re: On placing the blame (Matt Bishop)
  - At last, a way to reduce [net]news traffic (Jerry Aguirre via Matthew P Wiener)
  - Safety-Critical Software in the UK (Appendix B of ACARD report)

- Volume 4 Issue 15 (20 Nov 86)
  - IBM VM/SP SP Cracked (Jack Shaw)
  - On placing the blame AND Safety-Critical UK Software (Bjorn Freeman-Benson)
  - On placing the blame (Scot Wilcoxon)
  - Safety-Critical Software in the UK (Scott E. Preece)
  - Computer-based stock trading (from Discover)
  - FAA's Role in Developing a Mid-Air Collision-Avoidance System (Chuck Youman)

- Volume 4 Issue 16 (22 Nov 86)
  - Banking machine almost ruins love life of Vancouver couple (Mark Brader)
  - 2+2= ? (Risks of self-testing, especially with nonexistent tests) (Lindsay)
  - Re: Computer-based stock trading (Roger Mann)
  - Re: appendix to ACARD report (Nancy Leveson)
  - Some further thoughts on the UK software-certification proposals (Dave Platt)
  - Dependable Computing and the ACM Communications (PGN)

- Volume 4 Issue 17 (24 Nov 86)
  - Computer Risks and the Audi 5000 (Howard Israel with excerpts from Brint Cooper, Charlie Hurd, Clive Dawson)
  - Risks of changing Air Traffic Control software? (Greg Earle)
  - Re: the UK Software-Verification Proposal (Bard Bloom)
  - Program Trading (Howard Israel, Eric Nickell, dmc)
  - Decision Making (Clive Dawson)

- Volume 4 Issue 18 (26 Nov 86)
  - RISKS, computer-relevance, where-to-place-the-blame, etc. (PGN)
  - Verification and the UK proposal (Jim Horning)
  - When the going gets tough, the tough use the phone... (Jerry Leichter)
  - Re: 60 minutes reporting on the Audi 5000 (Eugene Miya)
  - Minireviews of Challenger article and computerized-roulette book (Martin Minow)
  - More on the UK Software-Verification Proposal (Bill Janssen)
- Volume 4 Issue 19 (26 Nov 86)
  - Very Brief Comments on the Current Issues (Kim Collins)
  - The Audi discussion is relevant (Hal Murray)
  - Audi 5000 (Roy Smith)
  - Laser-printer health risks; also, how to get ACARD report (Jonathan Bowen)
  - Data point on error rate in large systems (Hal Murray)
  - Re: Program Trading (Roger Mann)
  - Technical merits of SDI (from Richard Scribner)
- Volume 4 Issue 20 (30 Nov 86)
  - Smart metals (Steven H. Gutfreund)
  - Risks of having -- or not having -- records of telephone calls
  - Audi and 60 Minutes (Mark S. Brader)
  - Audi 5000/Micros in cars and the Mazda RX7 (Peter Stokes)
  - Automated trading (Scott Dorsey)
  - "Borrowed" Canadian tax records; Security of medical records (Mark S. Brader)
- Volume 4 Issue 21 (30 Nov 86)
  - Risks of Computer Modeling and Related Subjects (Mike Williams--LONG MESSAGE)
- Volume 4 Issue 22 (2 Dec 86)
  - More Air Traffic Control Near-Collisions (PGN)
  - Re: satellite interference (Jerome H. Saltzer)
  - "Welcome to the .... system": An invitation? (Bruce N. Baker)
  - Replicability; econometrics (Charles Hedrick)
  - Re: Risks of computer modeling (John Gilmore)
  - Computerized weather models (Amos Shapir)
  - Active control of skyscrapers (Warwick Bolam)
  - Privacy in the office (Paul Czarnecki)
  - Kremlin is purging dimwitted scientists (Matthew P Wiener; also in ARMS-D)
- Volume 4 Issue 23 (3 Dec 86)
  - The persistence of memory [and customs officials] (Richard V. Clayton)
  - America's Cup - floppies held to ransom (Computing Australia via Derek)
  - Some thoughts regarding recent postings: blame and causality (Eugene Miya)
  - Microcomputer controlled cars (not Audi) (Miriam Nadel)
  - Re: Welcome to the system (Ronda Henning)
  - Re: Automated trading (Scott Dorsey)
  - Active control of skyscrapers (Herb Lin)
- Volume 4 Issue 24 (5 Dec 86)
  - Criminal Encryption & Long Term effects (Baxter)

- Criminals and encryption (Phil Karn)
- Re: ATC Near-Collisions (Rony Shapiro)
- High Availability Systems (PGN)
- Plug-compatible modules (PGN)
- "Satellite interference" (Lauren Weinstein)
- Re: Privacy in the office (Brint Cooper)
- ACARD Report (Samuel B. Bassett)

Volume 4 Issue 25 (7 Dec 86)

- Child electrocuted (Anonymous, Brad Davis, Paul Nelson) [READ ALL 3!]
- On models, publications, and credibility (Bob Estell)
- Encryption and criminals (Perry Metzger, Fred Hapgood)
- Mode-C altitude transponders (Dan Nelson)
- ATM Limits (Richard Outerbridge)
- Taking the 5th (Jerry Leichter)

Volume 4 Issue 26 (10 Dec 86)

- Computer Error Endangers Hardware (Nancy I. Garman)
- "One of the Worst Days Ever for Muni Metro, BART" (PGN)
- Korean Air Lines Flight 007 (Steve Jong)
- Plug Compatible Modules; Criminal Encryption (David Fetrow)
- More on skyscraper control (Mike Ekberg)
- Satellite interference (James D. Carlson)
- (Il)legal Encryption (Richard Outerbridge)
- Software article in _Computer Design_ (Walt Thode)
- Heavy metal and light algorithms (PGN)
- Suit against Lotus dropped (Bill Sommerfeld)

Volume 4 Issue 27 (11 Dec 86)

- Computerised Discrimination (Brian Randell)
- Belgian Paper transcends computer breakdown (Martin Minow)
- Re: Plug-compatible modules (Keith F. Lynch)
- Re: Criminal Encryption (Keith F. Lynch, Ira D. Baxter, Dave Platt)
- Re: More on skyscraper control (Brint Cooper)
- The Second Labor of Hercules (Dave Benson)

Volume 4 Issue 28 (12 Dec 86)

- Mount a scratch giraffe, too? Make that several. (Jim Horning)
- Elf debuts as parking attendant (Kevin B. Kenny)
- Plug-compatible plugs (Chris Koenigsberg, Henry Schaffer)
- An Amusing Article on the Taxonomy of "Bugs" (Lindsay F. Marshall)
- Satellite interference (Lauren Weinstein)
- Fast-food computers (Scott Guthery)
- Re: More on skyscraper control (Chuck Kennedy)
- Re: Risks of Computer Modeling (Craig Paxton)
- Re: Computerized Discrimination (Randall Davis)
- Computers and Educational Decrepitude (Geof Cooper)
- Symposium -- Directions and Implications of Advanced Computing (Jon Jacky)

Volume 4 Issue 29 (14 Dec 86)

- America's Cup: Left-over Digital Filter (Bruce Wampler)

- Some additions to the "bug" taxonomy (Dick King)
- Re: uninterruptible power (Ted Lee)
- Trade-offs between BMD architecture and software tractability (Herb Lin)
- Re: Criminal encryption (Garry Wiegand)
- Computerised Discrimination (Scott Preece)
- More on Incompatible Plug-Compatible Monitors (Al Stangenberger)

Volume 4 Issue 30 (16 Dec 86)

- Arpanet outage (Andrew Malis)
- Dynamic Signature Verification (Robert Stroud [and Brian Randell])
- Wobbly skyscrapers and passive vs. active controls (Niall Mansfield)
- Re: The Audi 5000 problems (Matt Smiley)
- Modifying bank cards (Rodney Hoffman)
- Credit card mag strips (Ted Marshall)
- Fast-Food Computing (Edward Vielmetti)
- "bugs" (Doug McIlroy, Jonathan Clark, Bob Estell)

Volume 4 Issue 31 (17 Dec 86)

- Don't sit too close! ("And Now, Exploding Computers") (Jerry Leichter)
- Car-stress syndrome (Robert D. Houk)
- Korean Air Lines Flight 007 (Niall Mansfield)
- Heisenbugs (Rob Austein [an example], Doug Landauer)
- Criminal Encryption (Bill Gunshannon [counterexample?])
- Taking the "con" out of econometrics... correction and a plea (Mike Williams)

Volume 4 Issue 32 (18 Dec 86)

- EXTRA! British Telecom payphone Phonecard broken?

Volume 4 Issue 33 (21 Dec 86)

- Help British Telecom save a WORM. (Scot E. Wilcoxon)
- Security of magnetic-stripe cards (Brian Reid)
- Korean Air Lines Flight 007 (Dick King)
- Car-stress syndrome (Dick King)
- Bugs called cockroaches [A True Fable For Our Times] (anonymous)
- Re: More on car computers (not Audi) (Miriam Nadel)
- Runaway Audi 5000 (John O. Rutemiller)

Volume 4 Issue 34 (23 Dec 86)

- Debit cards that don't (Edward M. Embick, PGN)
- Re: security of magnetic-stripe cards (Henry Spencer)
- Plug-compatible plugs (Henry Spencer)
- Runaway Audi 5000 (Mark Brader)
- Ozone layer (Mark Brader)
- Another heisenbug (Zhahai Stewart)
- More "bugs" (Tom Parmenter via Richard Lamson)
- Computer Malpractice (Dave Platt)
- Financial Servomechanisms (Brian Randell)

Volume 4 Issue 35 (3 Jan 87)

- Computer Gets Stage Fright (Chuck Youman)
- Still More on PhoneCards (PGN)

- Miscarriages Up in Women Exposed In Computer-Chip Process (Martin Minow)
- Across the Atlantic with Cast Iron (Earl Boebert)
- Heisenbugs -- Two more examples (Maj. Doug Hardie)
- Risks Involved in Campus Network-building (Rich Kulawiec)
- Update on Swedish Vulnerability Board Report (Martin Minow)
- DES cracked? (Dave Platt)

Volume 4 Issue 36 (6 Jan 87)

- A Heisenbug Example from the SIFT Computer (Jack Goldberg)
- More Heisen-debugs (Don Lindsay)
- The Conrail train wreck (PGN)
- Software glitches in high-tech defense systems (from Michael Melliar-Smith)
- Computer program zeroes out fifth grader; Computerized gift-wrap (Ed Reid)
- Videocypher, DES (Jerry Leichter)
- More on the possible DES crack (David Platt)
- Campus LANs (James D. Carlson, Don Wegeng, Henry Spencer)
- Engineering Ethics (Chuck Youman)

Volume 4 Issue 37 (7 Jan 87)

- Re: vulnerability of campus LANs (Ted Lee, David Fetrow)
- Re: DES cracked? (Henry Spencer)
- Cellular risks (from Geoff Goodfellow via PGN)
- "Letters From a Deadman" (Rodney Hoffman)
- Stock Market Volatility (Randall Davis)
- Engineering ethics (Dick Karpinski)
- Computerized Discrimination (Ken Laws)

Volume 4 Issue 38 (8 Jan 87)

- As the year turns ... (Jeffrey Mogul)
- Automobile micros (Hal Murray)
- Chemicals in semiconductor manufacturing (Michael Scott)
- Cellular -- Ref to Geoff (via PGN)
- "Misinformation"?? (Dick Karpinski)
- Burnham Book -- A Recommendation (Alan Wexelblat)
- Engineering Ethics (Dan Ball)
- Re: Stock Market Volatility (Richard A. Cowan)

Volume 4 Issue 39 (11 Jan 87)

- Re: As the year turns ... (Jerry Saltzer)
- 911 computer failure (PGN)
- Engineering tradeoffs and ethics (Andy Freeman, Ken Laws, George Erhart)
- Re: computerized discrimination (Randall Davis)

Volume 4 Issue 40 (14 Jan 87)

- Phone Cards (Brian Randell)
- It's No Joke!! (Microwave oven bakes 3 yrs of PC data) (Lindsay Marshall)
- Automation bottoms out (PGN)
- Amtrak train crash with Conrail freight locomotive -- more (PGN)
- Re: Cellular risks (Robert Frankston)
- Re: Ask not for whom the chimes tinkle (Tom Perrine via Kurt Sauer)
- Re: Engineering ethics (PGN)

- Repetitive Strain Injury and VDTs (Mark Jackson)
- Safety Officers and "Oversight" (Henry Spencer)

Volume 4 Issue 41 (19 Jan 87)

- Audi 5000 recall (Dave Platt)
- UK EFT Risks (Brian Randell)
- Another Bank Card Horror Story (Dave Wortman)
- Stock Market behavior (Rob Horn)

Volume 4 Issue 42 (23 Jan 87)

- A scary tale--Sperry avionics module testing bites the dust? (Nancy Leveson)
- Computer gotcha (Dave Emery)
- Re: Another Bank Card Horror Story (Robert Frankston)
- Stock Market behavior (Howard Israel, Gary Kremen)
- Engineering models applied to systems (Alan Wexelblat)
- Re: British EFT note (Alan Wexelblat)
- Train Wreck Inquiry (Risks 2.9) (Matthew Kruk)
- Cost-benefit analyses and automobile recalls (John Chambers)

Volume 4 Issue 43 (26 Jan 87)

- "Cable `Hackers' Claim Scrambler is History"; other breaches (PGN)
- Re: VideoCypher II (Michael Grant)
- Re: DES cracked? (Douglas Humphrey)
- Re: Billions (Brian Randell)
- GM On-Board Computers (Wes Williams)
- Active control of skyscrapers (Peter G. Capek)

Volume 4 Issue 44 (29 Jan 87)

- Air Traffic Control -- More Mid-Air Collisions and Prevention (PGN)
- Time warp for Honeywell CP-6 sites (P. Higgins)
- GM On-Board Computers (Martin Harriman)
- Loose coupling (Ephraim Vishniac)
- Units RISKS and also a book to read (Lindsay F. Marshall)
- Re: Unit conversion errors (Alan M. Marcum, Keith F. Lynch)
- DP Ethics: The "Stanley House" Criteria (Pete McVay)

Volume 4 Issue 45 (2 Feb 87)

- DATE-86, or The Ghost of Tinkles Past (Rob Austein)
- Computerised Discrimination (an update) (Brian Randell)
- Another non-malfunctioning alarm (Jeffrey Thomas)
- Re: Engineering models applied to systems, RISKS-4.42 (Joseph S. D. Yao)
- Re: A scary tale--Sperry avionics module testing bites the dust? (D.W. James)

Volume 4 Issue 46 (9 Feb 87)

- TV-program on PBS: NOVA - Why Planes Crash (Werner Uhrig, Michael Harris)
- Electronic steering (Steve McLafferty)
- Senior to Repay Bank 25,000 Dollars (Steve Thompson)
- Recursive risks in computer design (McCullough)
- Library Failure (Chuck Weinstock)
- CP-6 time warp update (the true story) (John Joseph via Paul Higgins)
- Glitch in the Computers and Society Digest mailing list... (Dave Taylor)

- More on British Phone fraud (Will Martin)
- Wall Street Journal article on Risks (Jerome H. Saltzer)

Volume 4 Issue 47 (16 Feb 87)

- The fielding is mutuel! (PGN)
- Another worm story (Dave Platt)
- Re: The student's extra $25,000 (Ronald J Wanttaja)
- Problems with the B-1B Bomber (Bill McGarry)
- Super-Smart Cards Are Here. (Leo Schwab)
- Iranamok Computer-Databased (Craig Milo Rogers)
- Re: electronic steering (Tom Adams, Amos Shapir)
- Re: Nova: Why Planes Crash (Alan M. Marcum)
- Re: Library computerization (Will Martin)
- Second British Telecom Fraud (Lindsay F. Marshall)

Volume 4 Issue 48 (18 Feb 87)

- Four near air misses in 1986; Radar failure (Lindsay F. Marshall)
- Computer failure causes flight delays (Rodney Hoffman)
- Real RISKS (as opposed to virtual risks) of aircraft (Eugene Miya)
- Trojan Horse alert (Al Stangenberger)
- Computerized Town Data Vanish (Jerry Leichter)
- Re: UCSD work on human error (Alexander Glockner)
- Connector risk (Rob Horn)
- Re: Electronic steering (Brint Cooper)

Volume 4 Issue 49 (22 Feb 87)

- A misplaced report (Danny Cohen)
- Relevance (Amos Shapir)
- Re: London ATC (Jonathan Clark)
- Disk space cleanup causes problems with on-line Bar Admission exam (David Sherman)
- Automatic Call Tracing for Emergency Services (Mark Jackson)
- Re: The student's extra $25,000 (Kee Hinckley)
- Re: Electronic steering (Hien B. Tang)
- Re: TV-program on PBS: NOVA - Why Planes Crash (Henry Spencer)
- Re: RJ (phone) connectors for terminals (Jordan Brown)

Volume 4 Issue 50 (23 Feb 87)

- Principles of RISKS (James H. Coombs)
- "Demon computer" (PGN)
- NSA Risks (Alan Wexelblat)
- Results of a recent security review (Mary Holstege)
- Electronic steering (Kevin J. Belles, Rick Sidwell, Kevin Oliveau, Mark L. Lambert)

Volume 4 Issue 51 (25 Feb 87)

- HiTech version of NixonTapes (Pete Lee)
- Re: Automatic Call Tracing for Emergency Services (Lee Naish)
- Air Traffic Control, Auto-Land (Matthew Machlis)
- Electronic steering (Spencer W. Thomas, excerpt from William Swan)
- Hurricane Iwa and the Hawaii blackout of 1984 (James Burke via Matthew P Wiener)
- Summary of a Talk by SANFORD (SANDY) SHERIZEN on Computer Crime (Eugene Miya)

Volume 4 Issue 52 (26 Feb 87)

- B-1 plagued by problems (PGN)
- Computer loses bus (Mark Biggar)
- Human errors (Brian Randell)
- Possessed terminal? (pom)
- Entertainment risks (Walt Thode)
- Automatic Call Tracing for Emergency Services (James Roche, Charley Wingate)
- "Active" car suspensions (Graeme Dixon)
- Altitude-Detecting Radar (Matthew Machlis)
- Re: Results of a recent security review (Andrew Klossner)
- Re: Sherizen talk; auto-landing (Eugene Miya)
- Air Traffic Control, Auto-Land (Scott E. Preece)
- Risks of autopilots (and risks of solutions) (Bill Janssen)
- Another difference between electronic control in cars and fighters (Brent Chapman)
- Re: Hurricane Iwa (Scott Dorsey)

Volume 4 Issue 53 (1 Mar 87)

- Setuid Patent (Lindsay F. Marshall)
- On PGN's editorial comment on human misuse of computers (Eugene Miya)
- An aside on the B-1 (Eugene Miya)
- Autolander discussion (Nancy Leveson)
- Re: Air Traffic Control, Auto-Land (Dean Pentcheff)
- Electronic Steering (Ray Chen, Herb Lin)

Volume 4 Issue 54 (2 Mar 87)

- Rockford Illinois Destroyed by Computer! (Chuck Weinstock)
- Ma Bell's Daughter Does Dallas (PGN)
- FAA Does Houston (PGN)
- Tempest Puget, or The Sound and the Ferries (PGN)
- Re: proper use of suid (Jef Poskanzer)
- Process Control (Chuck Weinstock)
- Risks in switching to computerized `people meters' (Bill Janssen)
- A lovely algorithm (Lindsay)

Volume 4 Issue 55 (3 Mar 87)

- Air Cargo system in chaos (Lindsay F. Marshall)
- ATM Cards Devoured (again!); Royal Shakedowne for Tickets (Robert Stroud)
- Re: Risks in the NSC computer archives (Carlton Hommel)
- Re: A Scary Tale--Sperry Avionics ... (Kevin Driscoll)
- Re: Altitude encoders: $1500 for Mode C? No, $750. (Jordan Brown)
- One more on fly/steer-by-wire (Jonathan Clark)
- Steer-by-wire cars (Doug Rudoff)
- Software Safety in ACM Computing Surveys (Daniel S. Conde)
- Computerized `people meters' for TV audience ratings (Niall Mansfield)
- More on Dallas Phone outage (Mark Linnig)
- Soliciting suggestions for 1988 CSC panel on liability (Gene Spofford)
- Conference on computing and society in Seattle -- REMINDER (Jon Jacky)

Volume 4 Issue 56 (5 Mar 87)

- Computer problems produce false weather warnings (Mike Linnig)
- Some postscript notes about Hurricane Iwa (Bob Cunningham)
- Tempest Puget (Bill Roman)

- Computer Aided Dispatching (James Roche)
- Teflon flywheels and safe software (Hal Guthery)
- Autoland and Conflict Alert (Alan M. Marcum)
- Re: Air Traffic Control, Auto-Land (Amos Shapir)
- Re: An aside on the B-1 (Henry Spencer)
- Plane Crashes (David Purdue)
- In defense of drive-by-wire (Mike McLaughlin)

Volume 4 Issue 57 (6 Mar 87)

- Re: Air Traffic Control, Auto-Land (David Redell)
- 911, drive-fly by wire, risks, and the American work ethic (Wes Williams)
- Re: drive by wire (Bennett Todd)
- Autoland (Peter Ladkin)
- Re: Puget Sound Ferry Boats (Bjorn Freeman-Benson)
- Credit Card Limits (Clive Dawson)
- NSA Monitored McFarlane House, Magazine Reports (Don Hopkins)

Volume 4 Issue 58 (8 Mar 87)

- The Sperry Plan, FAA Certification, and N-Version Programming (Nancy Leveson)

Volume 4 Issue 59 (8 Mar 87)

- Safe software (Geraint Jones)
- Computer Problem causes airline financial loss (Rob Horn)
- Re: Altitude Encoders... expensive for some (Ronald J Wanttaja)
- Influence of goal selection on safety (Henry Spencer)
- Re: Puget Sound Ferry Boats (Dennis Anderson, Robert Frankston, Bjorn Freeman-Benson)
- GOES satellites, Scotchbrite, Gnomic Maxims, and Mr. Bill (Martin Harriman)
- Spreadsheet budget helping legislators (Scot E. Wilcoxon)

Volume 4 Issue 60 (9 Mar 87)

- Feel better now? (Martin Minow) [Risk probabilities in nuclear power]
- Computers in the Arts (or The Show Must Go On ...) (Jeannette Wing)
- Sensitive Intelligence Document Published On Magazine Cover(Stevan Milunovic)
- Mode-C Transponders (Phil R. Karn)
- Physical risks and software risks (Eugene Miya)
- Safe software (Scott E. Preece)
- Helicopter rotor failures (Peter Ladkin)
- Re: Electronic steering (D. V. W. James)
- Altitude Encoders... expensive for some (Herb Lin)
- F-104 (Elliott S. Frank)

Volume 4 Issue 61 (10 Mar 87)

- More on human errors (Brian Randell)
- Re: Teflon flywheels and safe software (Brian Randell)
- Re: Computers in the Arts (Alan Wexelblat, Jeffrey R Kell)
- Local telephone service problems (Jonathan Thornburg)
- Computer Failure Delays Flights at Atlanta Airport (PGN)
- Ozone hole a false alarm? (Henry Spencer)
- More on Requiring Mode C transponders (John Allred, Ken Calvert)

Volume 4 Issue 62 (11 Mar 87)

- "Software Safety: What, Why, and How" (Minireview by Jim Horning)
- Beef with Restaurant's Hi-Tech Computer (Yigal Arens)
- Electronic Steering (Mike Brown)
- Enhanced 911 risks (Mike Brown)
- Computers in the arts (Don Craig, Glenn Trewitt)
- Mode C (Ken Calvert)
- Re: Plane Crashes (Ronald J Wanttaja)
- Re: Results of a recent security review (Arnold D. Robbins)
- Risks of Maintaining RISKS -- and a reminder for BITNET readers (PGN)

Volume 4 Issue 63 (12 Mar 87)

- Re: Teflon flywheels and safe software (Al Mok)
- Re: Electronic Steering (Bob Ayers)
- Inputs For Quantitative Risk Assessment (Hal Guthery)
- Re: Active car suspension (Geof Cooper)
- Ozone hole a false alarm? (Mark Brader)
- Phone problems (RISKs in auto-dialers) (David Barto)
- Re: Mode C Transponders (Jan Wolitzky)
- Automatic Landing Systems (Hugh LaMaster)
- F-111 Losses (Rob Fowler)
- Re: Computers in the Arts (Computer lighting) (Shannon Nelson)

Volume 4 Issue 64 (16 Mar 87)

- Computer-lighting board nearly causes WWIII (Brent Laminack)
- Computerized telephone sales pitch meets emergency broadcast number (Brent Laminack)
- Furniture risks -- Vanishing Diskettes (Lee Breisacher)
- Reprise on the UK Government's ACARD Report (Brian Randell)
- Last minute changes (Roy Smith)
- Risk in ``High'' Financing (Michael Wester)
- Risk at Crown Books (Scott R. Turner)
- Human errors in computer systems -- another reference (Jack Goldberg)
- Requests for War Stories in Scientific Programming (Dennis Stevenson)
- TFR and F-111s (Eugene Miya)
- An Open University Text Book (Brian Randell)
- US NEWS article on 'Smart' Weapons - questions and concerns (Jon Jacky)

Volume 4 Issue 65 (19 Mar 87)

- Largest computer crime loss in history? (Gary Kremen)
- Health hazards of poorly placed CRT screens (Gregory Sandell)
- Re: Computerized telephone sales pitch ... (Robert Frankston)
- Re: phone key-pad speed vs accuracy (Andrew Klossner)
- ATM experience (Joe Herman)
- Computerized Telemarketing (Rob Aitken)
- Submission impossible? (PGN)
- Risk at Crown Books (Christopher Garrigues)
- Altitude Encoders... expensive for some (Herb Lin)
- RTD Ghost Story: a Phantom Warehouse (Eric Nickell)

Volume 4 Issue 66 (22 Mar 87)

- Question for Risks Readers on Overcoming Information Overload with Technology (Dave Taylor)
- Fumes from PC's (Lauren Weinstein)
- Re: health hazards of poorly placed CRT screens (Brinton Cooper)

- How to lose your ATM card (Jan Kok)
- Re: ATM experience (Bruce McKenney)
- Re: Increased Telephone Switching Capabilities (Dan Graifer)
- Releasing the phone line (edg)
- Automatic dialing devices in Canada (Michael Wagner)
- Overconfidence in Airplane Computers? (Ted Lee)

Volume 4 Issue 67 (24 Mar 87)

- Winch is the greatest risk in a theater? (Dave Wortman)
- DC9 Computer Failure (Earl Boebert)
- Health hazards associated with VDU use: eyestrain (John J. Mackin)
- Who called? (Jerome M Lang)
- Car Phone Intercept -- implications of captured data (Alex Dickinson)
- Re: Increased Telephone Switching Capabilities (Michael Wagner)
- Re: Telephone switches (Bjorn Freeman-Benson)
- Re: ATM experience (Roy Smith)
- Risks of ATM machines (Mike Linnig)
- Bank troubles, M.E. magazine (David Chase)
- Re: "The Choking Doberman..." (Elliott S. Frank)
- Newspaper article on Audi 5000S (Mark Brader)

Volume 4 Issue 68 (26 Mar 87)

- Re: Health hazards associated with VDU use: eyestrain (Barry Gold) ... and fluorescents (Re: RISKS-4.67) (Brad Davis) ... and related injuries (Jeremy Grodberg)
- Conference on Computers and Law (David G. Cantor)
- Re: runaway motors (Don Lindsay)
- The social implications of inadvertent broadcasts (Donn Seeley)
- Re: Increased Telephone Switching Capabilities (Andrew Klossner)
- Re: phone number of caller (Don Lindsay, Jeremy Grodberg)
- Hang-ups (Paul Wilcox-Baker)

Volume 4 Issue 69 (27 Mar 87)

- Cellular phone fraud busts (thanks to Geoff Goodfellow)
- "... and its fate is still unlearned..."; robotic exploration of Mars (Martin Minow)
- Re: Returned mail -- "Host unknown" (Richard Schedler and PGN)
- Re: Phone problems (Larry E. Kollar)
- Re: ATM experience (Brent Chapman)

Volume 4 Issue 70 (1 Apr 87)

- Rocket Shot Down By Faulty ``Star Wars'' Weapon (Phil R. Karn)
- ATMs, phones, health hazards, and other sundry subjects (PGN)
- Computer Risks in Theatre (Warwick Bolam)
- PC fumes (Dick King)
- A real eye-catching headline (David Chase)
- Risks of being fuzzy-minded (Ted Lee)
- ATM discussions (gins)
- Re: ATM experience ... it actually gets worse (Allen Brown)

Volume 4 Issue 71 (5 Apr 87)

- Re: A real eye-catching headline -- nuclear safety (Jerry Saltzer, Peter G. Neumann, Henry Spencer)
- A non-fail-safe ATM failure (Don Chiasson)

- Fumes from computers and other electronic appliances (Richard Thomsen)
- Open University Fire (Lindsay F. Marshall)

Volume 4 Issue 72 (8 Apr 87)

- New kind of computer-technology-related deaths? (PGN)
- Conrail Sale Funds Transfer (Chuck Weinstock)
- Re: "Inherently safe nuclear reactors" (Phil Ngai)
- A different RISK? (in-flight control computers) (Peter Ladkin)
- Fumes from computers and other electronic appliances (Mark W. Eichin)
- VDT related skin cancer? (Chris Koenigsberg)

Volume 4 Issue 73 (11 Apr 87)

- Unintentional information dissemination (George W. Dinolt)
- Computers & Personal Privacy (Steve Thompson)
- Air Traffic Control in the UK (Lindsay F. Marshall)
- Air Traffic Control in the USA (PGN)
- Re: "Inherently safe nuclear reactors" (Jim Carter)
- Submarine reactor safety (Jim Hunt)
- Re: A different RISK? (in-flight control computers) (Ronald J Wanttaja)
- Risks"-taking" of in-flight control computers (Eugene Miya)
- Software Risks with Cable TV (Walt Thode)
- The UNIX rwall problem ["My Broadcast"] (Jordan K. Hubbard)

Volume 4 Issue 74 (14 Apr 87)

- Re: In-flight control computers (Henry Spencer)
- Trojan Horse alert (Al Stangenberger)
- The Limits of Software Reliability (Brian Randell)
- Re: Conrail Sale Funds Transfer -- and a 747 overflow (Henry Spencer)
- Re: VDT related skin cancer? (Henry Spencer)
- Re: Open University Fire (Henry Spencer)
- DES Second Review Notice [on the RISKS OF STANDARDS] (David M. Balenson)
- Bank Computers (Not ATM's) (Ken Ross)
- The Marconi Affair (Brian Randell)

Volume 4 Issue 75 (22 Apr 87)

- Flight control risks (Peter Ladkin)
- ``More on risky high-g piloting'' (Tom Perrine)
- Checklist stops risks? (Joseph Beckman)
- Radiation risk at airports? (Paul Stewart)
- How to post a fake (Chuq Von Rospach, Rob Robertson)
- Re: Bank Computers (Not ATMs) (Kuhn)
- Correction to Conrail Sale Funds Transfer (Mark Brader)
- "Reliability Theory Applied to Software Testing" (HP Journal)(Rich Rosenbaum)

Volume 4 Issue 76 (22 Apr 87)

- Risks of Warranties (Jim Horning)
- Re: Checklist stops risks? (Jerome H. Saltzer)
- Newer highly maneuverable planes on board and checklists (Eugene Miya)
- Aircraft risks (Peter Ladkin)
- Neutron beam detection (Scott Dorsey)

Volume 4 Issue 77 (23 Apr 87)

- 'Hackers' hit the Jackpot (Michael Bednarek)
- Fidelity Mutual Funds Money Line feature (Chris Salander via Barry Shein)
- VCRs, Telephones, and Toasters (Martin Ewing)
- Checklists, Aircraft risks, and Neutrons (Eugene Miya)
- Neutron Beams for Explosives Detection (Marco Barbarisi)
- Forgery on Usenet (Brad Templeton)
- Re: How to post a fake (Wayne Throop)

Volume 4 Issue 78 (26 Apr 87)

- Re: Fidelity Mutual Funds Money Line feature (Martin Ewing, Brint Cooper)
- Re: Forgery on Usenet (Matt Bishop)
- Re: VCRs, Telephones, and Toasters (Mark Jackson)
- References on computer-professional certification (John Shore)
- CPSR/Boston presentation: "Reliability and Risk"

Volume 4 Issue 79 (2 May 87)

- Risks of RISKS resurgent -- CSL DEAD FOR THREE DAYS, STILL HALF DEAD
- Re: Fidelity Mutual Funds Money Line feature (Amos Shapir)
- Wheels up (Martin Minow)
- Special Risk Assessment issue of 'Science' (Rodney Hoffman)
- Radiation hazards to computers (Wm Brown III)
- Neutron beam detection (Richard H. Lathrop)
- Computer Database Blackmail by Telephone (Steve Summit)
- Liability Law in the UK (Brian Randell)

Volume 4 Issue 80 (5 May 87)

- Computer Risks at the Department of Transportation (PGN)
- Computerized advertising network used to fence hot circuits (PGN)
- EPROMS and "Wimpy" Energy Physics (Patrick Powell)
- Re: Wheels up (Richard M. Geiger, Jerry Hollombe>
- Liability for software "unless you buy our method" (John Gilmore)

Volume 4 Issue 81 (7 May 87)

- Cadillac to recall 57,000 for computer problem (Chuq Von Rospach)
- Public E-Mail Risks? (Brian M. Clapper)
- Wheels up (and simulators) (Eugene Miya, Doug Faunt, Matt Jaffe)
- Subject: Re: the Marconi deaths (an update) (Brian Randell)

Volume 4 Issue 82 (10 May 87)

- Information Age Commission (PGN)
- Another computer taken hostage (Joe Morris)
- Larceny OF Computers, not BY Computers (Pete Kaiser)
- Risks of superconductivity (Eugene Miya)
- UK Liability Law (follow-up) (Brian Randell)

Volume 4 Issue 83 (12 May 87)

- Risks of sharing RISKS (Ted Lee)
- Information Commission (Jim Anderson)
- ``How a Computer Hacker Raided the Customs Service'' (Michael Melliar-Smith)
- Computer thefts (Jerry Saltzer)

- Bomb Detection by Nuclear Radiation (Michael Newbery)
- Computer floods summer course registration at U. of Central Florida (Mark Becker)
- A password-breaking program (Dean Pentcheff)
- Sidelight on the Marconi Deaths (Lindsay F. Marshall)
- Software Reliability book by Musa, Iannino and Okumoto (Dave Benson)
- "The Whistle Blower" (Jeff Mogul, via Jon Jacky)

Volume 4 Issue 84 (12 May 87)

- Re: Information Age Commission (Herb Lin, Richard Cowan, Bob Estell, David LaGrone, Michael Wagner)
- Re: Information Age Commission; Summer Courses at UCF (William Brown III)
- Re: A password-breaking program (Dean Pentcheff, Jerry Saltzer, Dave Curry)
- Re: Computer thefts (Michael Wagner)
- Re: Computer-related Cadillac recall (Jeffrey R Kell)

Volume 4 Issue 85 (14 May 87)

- Holiday reading (Jim Horning)
- Hey, buddy, wanna buy a phone call cheap? (PGN)
- Re: Information Age Commission (Ted Lee, SEG)
- Information Age Commission and the number of readers of RISKS (David Sherman)
- Lockable computers (Pat Hayes)
- How a Computer Hacker Raided the Customs Service -- Abstrisks (a nit) (Paul F Cudney)

Volume 4 Issue 86 (18 May 87)

- ATM Fraud (Chuck Weinstock)
- Between Iraq and a Hard Place [Protect Your Phalanx] (William D. Ricker)
- Wozniak Scholarship for Hackers (Martin Minow)
- Information Overload and Technology? (David Chess)
- Passwords, thefts (Andrew Burt)
- Passwords, sexual preference and statistical coincidence? (Robert W. Baldwin)

Volume 4 Issue 87 (20 May 87)

- Computer Libel: A New Legal Battlefield (PGN from Digital Review)
- Electric chair tested by car insurer (Bill Fisher from Machine Design)
- Computers and Open Meetings laws (Barbara Zanzig)
- Re: Phalanx (Chuck Weinstock)
- Choosing a password (Jonathan Bowen)
- Re: Passwords, thefts (Michael Wagner)
- Nuclear Plant Emergency Plan: In Event of Quake, Smash Toilets (UPI via Don Hopkins, Michael Grant, and Geoff Goodfellow)

Volume 4 Issue 88 (21 May 87)

- Re: Phalanx (Phil Ngai)
- Open meeting laws (Dave Parnas)
- Concerning UN*X (in)security (Mike Carlton)
- Ed Joyce, Software Bugs: A Matter of Life and Liability (Eugene Miya)
- Risks and system pre-login banners (PGN)
- Risks of Running RISKS, Cont'd. (PGN)

Volume 4 Issue 89 (24 May 87)

- Factory Robots Killing Humans, Japan Reports (PGN)
- Mysterious BART power outage (PGN)

- More on the Master Password attack (PGN)
- Measures, countermeasures, and under-the-countermeasures (PGN)
- Phalanx (Scott Dorsey, Henry Spencer)
- rhosts (Anthony A. Datri)
- Computer Bill of Rights (Eugene Miya)
- Credit Information Access (Ron Heiby)
- Open meeting laws (Jonathan Handel)
- Privacy and Email - The Law Takes Notice (Jerry Leichter)

Volume 4 Issue 90 (25 May 87)

- Laser guided missiles... (Herb Lin)
- Computer use costs civil servants $1,270 (Matthew Kruk)
- Liability in Expert Systems (David Chase)
- Electronic Communications Privacy Act (Dave Curry)
- ATM security (Kenton Abbott Hoover via Martin Minow)
- Communications Technology Aids Criminals (Larry Lippman)

Volume 4 Issue 91 (28 May 87)

- Electromagnetic Interference in Japan (Lindsay F. Marshall)
- Risk of Inappropriate Technology to Prevent Password Overwrite (Paul Stachour)
- Passwords and Statistics (Earl Boebert)
- Why Cellular phones at the Indy 500? (Robert Adams)
- Information Security Products and Services Catalog by NSA (Kurt F. Sauer)
- Re: TRW "Credentials" (John R. Levine) [Other messages overlapped, omitted]
- Phalanx Schmalanx (PGN, Mike Trout, Torkil Hammer)
- Laser guides (Jon A. Tankersley)
- Re: Risks of running Risks (Jeff Woolsey, Will Martin)
- Re: Computer thefts (David Phillip Oster)

Volume 4 Issue 92 (30 May 87)

- Computer matching of cats and dachshunds (Rick Kuhn)
- Electromagnetic Interference (EMI) & Liability (Richard S D'Ippolito)
- Horror story about inadvertent wiretapping (Gordon Davisson)
- ATM fraud (Bob Johnson)
- Computer thefts (Mike Alexander, Brint Cooper)
- Shooting Down Exocet Missiles (Mark S. Day)
- Phalanx is unreliable? (Lorenzo Strigini)
- Stark Incident (Eugene Miya)
- Technical error in item "Phalanx Schmalanx" (Mark Brader)
- Phalanx; Laser guides (Phil Ngai)
- Laser guided anti-tank weapons (Eugene Miya)
- Unfair testing (Paul Peters)
- "Credentials", Privacy, etc. (Willis Ware, Alan R. Katz)

Volume 4 Issue 93 (1 Jun 87)

- Soviet Air Defense Penetration (Martin Minow, Eugene Miya)
- Exocet, PHALANX, chaff, and missile defense (Sean Malloy)
- Re: Phalanx Schmalanx (Mike Iglesias)
- Re: Computer thefts (Brian Matthews)
- TRW's Credentials (Jonathan Handel)

Volume 4 Issue 94 (2 Jun 87)

- Australian Computer Crime (Donn Parker)
- PCs and Computer Fraud (PC Week via PGN)
- Technological vs. (?) human failure (Nancy Leveson)
- Risk of Inappropriate Technology to Prevent Password Overwrite(Henry Spencer)
- A twist on modems calling people (Steve Valentine)
- Risks of Compulsive Computer Use (Steve Thompson)
- Perhaps the Bill of Rights you sought? (Bruce Wisentaner)
- Error(s) in "Phalanx Schmalanx" (Mike Trout)

Volume 4 Issue 95 (3 Jun 87)

- COMPASS '87, of particular interest to the RISKS audience (Stan Rifkin)
- Re: Run-time checks (Jerome H. Saltzer)
- Risks of Inappropriate Technology to Prevent Password Overwrites (Michael Robinson)
- Clarification of PL/I array checking (Michael Wagner)
- Risks for computer junkies (Robert Hartman)
- Re: When Computers Ruled the Earth (Bank Stupidity) (Ed Sachs)
- Clarification on CHAPPARAL and VULCAN (Bill Gunshannon)

Volume 4 Issue 96 (6 Jun 87)

- Lightning Strikes Twice At NASA (Matthew P Wiener)
- Iraqi cockpit navigation system placed Stark in exclusion zone? (Jon Jacky)
- Run-time checks (Howard Sturgis, Henry Spencer, James M. Bodwin, Alan Wexelblat)
- Error Checking and Norton's Assembly Language Book (James H. Coombs)
- Re: Risks of Compulsive Computer Use (Douglas Jones)
- A reference on Information Overload; a Paradox of Software (Eugene Miya)
- Computerholics (James H. Coombs)
- Naval Warfare -- on possible non-detonation of missiles (Mike McLaughlin)

Full Body Scan and pat down in progress

You were warned....

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 7

Friday, 13 June 1986

Contents

- Eastport Study Group report ("Science" article)
  Pete Kaiser
- An additional SDI problem: sensor technology
  Jon Jacky
- Shuttle software and CACM
  James Tomayko [and Herb Lin]
- Privacy laws
  Bruce O'Neel
- A mini-editorial on running the RISKS Forum
  PGN
- Info on RISKS (comp.risks)

Eastport Study Group report ("Science" article)

Systems Consultant <kaiser%renko.DEC@decwrl.DEC.COM>
Thursday, 12 Jun 1986 04:54:22-PDT

"Science", in the issue of 9 May 1986, contains an article on "Resolving the Star Wars Software Dilemma". The subhead reads:

> A panel of computer scientists has concluded that computers will be able to manage a strategic defense system -- but only if battle management is designed in from the beginning.

More, from within the article:

> ...The report is in fact a scathing critique of the way the Pentagon handles high-technology weapons design in general and software development in particular. It deals with important questions about the limits of computing, the nature of reliability, the organization of large, complex systems, and the nature of strategic defense itself.
>
> And in a striking paradox, it validates what the program's many critics have been saying about the infeasibility of Star Wars software. ...

First, they say, battle management is tractable only if SDIO and its defense-industry contractors give up their tacit assumption that software is an "applique," something that can be sprinkled on preexisting weapons and sensors like pixie dust to turn them into a working defense system. This assumption was quite evident in SDIO's so-called "Phase I" architecture studies, which were completed in 1985 and which seemed to concentrate almost exclusively on hardware.

The "paradox", as I read the Study Group's findings in the article, is that although it might be possible to design systems that did effective battle management (in some interpretation of "effective") by integrating software and hardware from the earliest stages of design, there is no sign whatever that this could happen in the real world of military contractors and politics. Thus, in the report's view, it is effectively impossible to build workable Star Wars systems.

Recommended, but not comforting, reading. (The name of the full report from the Eastport Study Group is "Summer Study 1985: A Report to the Director of the Strategic Defense Initiative Organization", December 1985.)

---Pete
Kaiser%furilo.dec@decwrl.dec.com decwrl!furilo.dec.com!kaiser
DEC, 2 Iron Way (MRO3-3/G20), Marlboro MA 01752 617-467-4445

Re: An additional SDI problem: sensor technology

Jon Jacky <jon@uw-june.arpa>
Thu, 12 Jun 86 22:32:55 PDT

> (Eugene Miya writes:) ... Where there are various groups watchdogging
> computing, but the more hardware oriented, EE areas such as radar have
> fewer opposition elements.

Sensors and signal processing comprise a larger portion of the SDI effort than anything else, according to many reports.

The most informative comments I have heard were by Michael Gamble, a vice president (I think) at Boeing, and head of that company's 'Star Wars' research programs. He administers about half a billion dollars worth of contracts. In a talk to the Seattle chapter of the IEEE on Nov. 14, 1985, he noted that the total SDI budget requests for fiscal years 1985 through 1990 would total about $30 billion, broken down as follows: Sensors $13B, directed energy weapons $7B, kinetic energy weapons $7B, Battle Management $1B, Survivability $2B. Sensors comprise almost half the total. (I do not know whether these proportions are maintained in the somewhat reduced budgets that get approved.)

Gamble also explained why he thought missile defense was once again plausible, after being debunked in the early 70's. "What has changed since then?" he asked rhetorically, and gave five answers, three of which involved sensors: first, long wave infrared detectors and associated cooling systems, which permit small warheads to be seen agains the cold of space; second, "fly's eye" mosaic sensor techniques (like the ones used on the F-15 launched ASATS and in the 1984 "homing overlay experiment") -- these are said to "permit smaller apertures" (I didn't catch the significance of that); and third, low-medium power lasers for tracking, designation, and homing. The other two factors were long-life space systems and powerful onboard computing capabilities.

There is a large computing component in the sensor field: digital signal processing. However, this area is not so well known to computer science types. Boeings largest SDI contract - over $300M - is for the "Airborne Optical Adjunct," an infrared telescope and a lot of computers mounted in a 767 airliner, apparently for experiments in sensing and battle management for midcourse and terminal phase. Two of the systems people involved in this project gave a seminar at the UW computer science department last January. They mentioned that the signal processing was being handled by the sensor people and they just regarded it as a black box.

I can think of two reasons why this area has received relatively little attention. First, there were no galvanizingly absurd statements about sensors from relatively prominent SDI proponents - nothing like James Fletcher calling for "ten million lines of error-free code," or all that bizarre stuff in the Fletcher report and elsewhere about launching pop-up X-ray lasers under computer control. Second, there is a lot secrecy in the sensor area-- unlike battle management, where the important issues do not turn on classified material. Gamble noted that "there is not that much that is classified about SDI, except things like, 'How far can you see? How far do you have to see?'" Needless to say, talking in detail about sensors would reveal how much we know about Soviet warhead characteristics, how good our early warning systems really are, and so forth.

-Jonathan Jacky
University of Washington

Shuttle software and CACM

<James.Tomayko@sei.cmu.edu>
Thursday, 12 June 1986 09:04:12 EDT

As referenced in the recent RISKS, the CACM case study is a somewhat decent introduction to the Shuttle onboard software. However, I would like to warn readers that the case study editors interviewed IBM FSD personnel *only*, with no attempt to talk to the customer, NASA, or the users, the astronauts.

I was under contract with NASA for three years to do a study of its use of computers in space flight, and my interviews with crews and trainers provided a somewhat more critical view of the software. Also, it is useful to remember that the primary avionics software system documented in the CACM study runs on four computers. Last count was that there are something over 200 processors on the orbiter (Source: Jack Garman, Johnson Space Center).

So, please take the CACM articles with a grain of salt.

Jim Tomayko
Software Engineering Institute

P.S. To forestall some mail: The earliest NASA will release my Technical Report is late 1987.

[In addition, Herb Lin responded to David Smith, included here for emphasis:
"This issue of CACM *is* a pretty good review of shuttle software. On the other hand, you must remember that the interview was with the people who were in primary charge of the project. Thus, you would be rather unlikely to hear about problems and so on that remained unresolved. That claim doesn't diminish the value of the article, but it should prompt caution in accepting the general impression it gives that all was (or is) just fine... Herb"]

Privacy laws

Bruce O'Neel <ZWBEO%VPFVM.BITNET@WISCVM.WISC.EDU>
Thu, 12 Jun 86 10:55 EDT

In response to the House law about computer communications privacy, I believe that the following is correct. Right now, communications are protected if they are telephone, mail, and other "traditional" technologies. One can not "wiretap" you without a warrant. The current laws don't cover computer communications or car phones or other "new" communications technology. According to what I read in Wash. Post this bill would consider car phone communications, computer communications, and others the same as the mail and land based phone calls.

Bruce O'Neel <zwbeo@vpfvm.bitnet>

A mini-editorial on running the RISKS Forum

Peter G. Neumann <NEUMANN@SRI-CSL.ARPA>
Fri 13 Jun 86 00:25:40-PDT

Life is usually a delicate balance among many tradeoffs. Running RISKS is no different:

The subject of Risks to the Public in Computer Systems involves tradeoffs among technical, social, economic, and political factors. These are very hard to assess, because each person's basis for judgment is likely to be different. (All of these factors are relevant in the broad sense, although we generally try to focus on the technical issues.) Some risks are intrinsic in technology; the question is under what circumstances are they worthwhile -- and that involves all of the factors (and more).

If messages were too superficial or issues too infrequent, most of you would lose interest. If issues and/or messages were very long or too frequent, you would most likely be overwhelmed. (But I occasionally get requests for single-message mailings from BITNET subscribers [who have not yet discovered undigestifiers?], although that presents many difficulties.)

If I put too much of my time into RISKS, my other responsibilities may suffer. If I put too little time in, you may suffer.

If I turn down the threshold and accept contributions that violate the masthead requirements (relevancy, soundness, objectivity, coherence, etc.), we all suffer. If you contribute junk and I don't reject it, you and I suffer. If I turn up the threshold and reject many contributions, I defeat one of the main purposes of RISKS, which is to be an open forum.

If RISKS were to take itself too seriously, or alternatively to become too frivolous, that would be bad. [I try to keep my pun level down, but occasionally I may slip a little.])

So, thanks for sticking with us in this experiment in communication on a vital topic. Please complain to RISKS-Request or to me when you are really unhappy. It can only help. Peter

---

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 8

Sunday, 15 June 1986

Contents

- Challenger, SDI, and management risks
  Dick Dunn
- Re: Risks from inappropriate scale of energy technologies
  Chuck Ferguson
- Distributed versus centralized computer systems
  Peter G. Neumann
- Privacy legislation
  Michael Wagner
- Info on RISKS (comp.risks)

Challenger, SDI, and management risks

Dick Dunn <nbires!rcd@ucbvax.Berkeley.EDU>
Fri, 13 Jun 86 12:24:29 mdt

The Challenger failure has an implication for SDI that I've not seen discussed much. I regard the solid-rocket failure as primarily a management failure and only secondarily an engineering failure. Why? Because according to the Rogers group reports, there had been serious concern with the possibility of failure of the O-ring seals, but it got lost or suppressed along the way. Challenger's ill-fated launch was apparently made in spite of best engineering advice to the contrary.

How does this apply to SDI? I'll give a sketch; I hope that other people will add more. SDI is under fire from several places (substantial part of Congress, various public-interest groups, many influential technical people). It is therefore important for the supporters (willing and/or appointed) of SDI to present a convincing case that SDI can "do the job." There is tremendous pressure to justify SDI. Translate this into "there is tremendous pressure to argue the case that SDI can be built and can work--whether or not it really can." To be blunt, there is a tremendous incentive to cover up any potential inability to build an SDI system or any inadequacy once it is built. Of course, if the SDI system is built, and is used, and fails, there will be much more lost than seven lives and some megabucks of hardware. (I have a hard time typing the word "terabucks":-) There probably Wouldn't even be a presidentially-appointed blue-ribbon investigative committee...

The hard questions: Do we have a way to manage a project of the magnitude of SDI that will give us any halfway-reasonable assurance that the project will work? Is there any technique that can be applied to reward those who discover problems and punish those who cover them up, instead of the other way around?

(My own experience, unfortunately, tells me that these aren't really hard questions. Rather, they are questions which are easily answered "no!" The difficulty in managing any large project, particularly one which involves a lot of software, is legendary.)

In summary, I'm saying that Challenger failed not for technical reasons-- I believe that the technical problems are real but surmountable--but for managerial reasons. Further, I think that we need to talk about SDI feasibility in more than technical terms; we need to address whether we could manage the project even if all of the technical problems were surmountable. The answer is anything but a clear "yes".

Dick Dunn

> [From The New York Times, Sunday, 15 June 1986:
>
> New York - The ''Star Wars'' anti-missile plan has been seriously and extensively damaged by the Challenger disaster and other setbacks in the American space program, aerospace analysts say. Officials of the anti-missile defense program, formally called the Strategic Defense Initiative, deny any serious damage to the program, but aerospace experts say the problems within the space program have sent shock waves through research programs. ...]

Re: Risks from inappropriate scale of energy technologies

Chuck Ferguson - SCTC <CTFerguson@HI-MULTICS.ARPA>
Thu, 12 Jun 86 20:06 CDT

In RISKS-3.6, Michael Natkin states:

> The public has long been duped into the idea that centralized energy management has its best interest in mind as we develop ever increasing electrical capacity. But centralized reactors and other "hard" technologies are extremely susceptible to terrorist attack and other failures, as has been mentioned before.

Centralized power supplies may be "extremely susceptible" to terrorists but their susceptibility to failure is not as high as being claimed. It is true that the consequences of a failure might be great; however, for a large centralized power plant it is economical to expend greater resources to prevent their failure (e.g., redundancy) than for the components of a distributed system. Furthermore, I submit that all current power systems have some degree of distributed or redundant functionality to allow periodic maintenance shutdowns if for no other reason.

I further submit that there is a significant risk associated with distributed systems which is being ignored. Many such systems are themselves dangerous when poorly maintained or operated improperly. There are also hazards associated with storing combustible fuels near a dwelling or other populated area. Witness the following:

- o How many chimney fires have you heard about since the "energy crisis" began? A fireplace is a relatively low-tech device yet some people manage to make them dangerous.

- o Why is it that several houses burn down at the start of every cold season? An oil-fired furnace is a relatively low-tech item also, yet every year someone's gets choked with soot and catches fire.

- o Ever heard of a methane gas explosion in a sewer system? I recently heard an amusing story about a manure fire at a horse ranch - ten years worth of horse manure had been piled in one place until one day it spontaneously caught fire.

One would be surprised how much damage some people can do with low-tech alternative energy. To paraphrase one of the better known 'computer security experts' [emphasis added], "Terrorists can never compete with incompetents". I wonder whether more people lose their lives each year in the commercial production of power or in incidents similar to the above.

With respect to the public "being duped" - sounds like another conspiracy theory to me (yawn).

Chuck Ferguson, Honeywell, Inc., Secure Computing Technology Center

---

Distributed versus centralized computer systems

Peter G. Neumann <Neumann@SRI-CSL.ARPA>
Sun 15 Jun 86 22:32:04-PDT

Although Chuck's note does not seem as closely related to RISKS as some of his past contributions, it suggests various additional comments. A myth of distributed computing systems is that distribution avoids centralized vulnerabilities. WRONG! The 1980 ARPANET collapse gave us an example of an accidentally propagated virus that contaminated the entire network. The first shuttle synchronization problem is a further example.

By distributing what has to be trusted, there may be more vulnerabilities -- unless the distributed components are TOTALLY autonomous -- in which case we are not really talking about DISTRIBUTED systems, but rather SEPARATE systems. Security flaws in the systems and networks can result in transitive vulnerabilities, or permit global compromises by iteration.

Further, the point raised by Chuck regarding maintenance is an important one in distributed computer systems, especially if some of the distributed sites are remote. Well, then, you say, let the field engineers dial up the remote site. But then that path provides a monstrous new vulnerability. Then we get solutions like the remote backup scheme proposed a while back that gets special privileges... Also, remember the fundamental flaws in the standard two-phase commit protocols, three-clock algorithms, and so on. Once again it might be useful to consider truly robust algorithms such as interactive consistency and Byzantine agreement. However, for every more complex would-be technical solution there are often further technical problems introduced. For every assumption that things have gotten better there seem to be even grosser counterexamples and further vulnerabilities outside of the computer systems. Thus,

> It is folly to trust software and hardware if an end-run can bypass or compromise the trusted components. But it is also folly to assume that sabotage is significantly less dangerous just because a system is distributed. That may be true in certain cases, but not generally.

Peter [Please excuse me if I have repeated some things that I said in earlier RISKS in a different context.]

Privacy legislation (RISKS-3.6)

Michael Wagner <ubc-vision!utcs!wagner@seismo.CSS.GOV>
Sat, 14 Jun 86 11:26:37 edt

```
>A news clipping from this morning's "Los Angeles Times" (page 2, The News
>in Brief):
>
>  The House Judiciary Committee voted 34 to 0 for a bill seeking to
>  bring constitutional guarantees of the right to privacy into the
>  electronic age.  The legislation would extend laws that now protect
>  the privacy of the mails and land-line telephone conversations to also
>  cover electronic mail and some telephones that use radio waves.
```

Does anyone have any idea how the last part (radio telephones) could be legally supported in view of other legal freedoms? I thought that one was free to listen to any frequency one wished in the US (Canada too). You don't have to trespass to receive radio signals.

Contrast this with the mails. The privacy of the mails is supported by property laws. That is, you put your mail into a box which belongs to the post office. If anyone breaks into that box (or the van which picks up the mail, etc) they are breaking property laws. Similarly for land lines. One has to 'trespass' to tap a land line.

It seems to me that the legislators have 'extended' the laws over an abyss. Or have I missed something?

The relevancy to RISKS, of course, is that most people don't think about the technology that radio-telephones use. I'm sure most people assume "it's a phone - it's (relatively) safe". Not true, of course. In fact, some people have used their own handsets to make phone calls on other peoples phones!

Michael Wagner

[I do not recall having pointed out in this forum the ease with which the cellular phone schemes can be spoofed, e.g., getting someone else to pay for your calls. There is another security/integrity problem waiting to be exploited. PGN]

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 9

Friday, 20 June 1986

Contents

- Informing the Senate on SDI
  Jim Horning
- A medical risk of computers
  Karen R. Sollins
- Risks of VDTs
  Alan Wexelblat
- Minor addition on Risks of Distributed Energy
  Ted Lee
- [Additional material back-logged. Delay due to PGN travelling.]
- Info on RISKS (comp.risks)

Informing the Senate

Jim Horning <horning@src.DEC.COM>
Wed, 18 Jun 86 12:21:05 pdt

The information in this message is political, not technical. However, it concerns the process of supplying technical information to those who must make political judgements, so I believe that it is germane to Risks.

Recent news accounts have indicated that the outcome in the Senate of requests for increased SDI funding is very unclear. Senators are having to take positions on a matter they don't fully understand, and many of them would like to be better informed.

I was contacted by an aide to Senator Proxmire for information about how David Parnas's criticisms of SDI software are viewed in the professional community. General Abramson and the SDIO have had some success in spreading the message that David Parnas is an isolated crank who is not taken seriously by those who actually build software.

I was able to express my own opinion and concerns, but cannot speak credibly for the entire professional community. Pound-for-pound, Risks probably contains more people qualified to make an informed judgement on

this issue than any other group I know how to reach. Whatever your views,
I would urge you to take the time to write a letter expressing them to

Mr. Douglas Waller
Office of Senator William Proxmire
United States Senate
Washington, DC 20510

Based on my experience, you can expect your letter to receive personal attention, and to carry weight according to your credentials and the cogency of your arguments. (This is in sharp contrast to my experience writing to my own senators and congressman.)

In addition to stating your own views clearly, it would probably help to indicate how they relate to Parnas's criticisms and (if you have read it) to the Eastport Report. In my own letter, I also devoted a paragraph to sketching my credentials; I don't much care for such self-advertisement, but thought I should give a starting point for any checking they cared to do, and the reasons why I felt qualified to comment on reliability and on aerospace software.

Jim H.

---

a medical risk of computers

"Karen R. Sollins" <sollins@XX.LCS.MIT.EDU>
Fri, 20 Jun 1986 10:37 EDT

My particular concern in the story that follows is that the designers and programmers probably can't know ALL the conditions for which to check. We all know that complete testing of complex systems is impossible. All too often we are put into a position of trading risks and benefits, and at least the risks (as in this case) are not and cannot be known completely.

Of course, another difficult question here is who is responsible for what happened and what should be done about it. Clearly for those three patients involved and their families and friends no amount of placing responsibility, punishment, or compensation can make up for what was done to them.

Karen Sollins

____
MAN KILLED BY ACCIDENT WITH MEDICAL RADIATION
(excerpted from The Boston Globe, June 20, 1986, p. 1)
by Richard Saltos, Globe Staff

A series of accidental radiation overdoses from identical cancer therapy machines in Texas and Georgia has left one person dead and two others with deep burns and partial paralysis, according to federal investigators.

Evidently caused by a flaw in the computer program controlling the highly automated devices, the overdoses - unreported until now - are believed to be the worst medical radiation accidents to date.

The malfunctions occurred once last year and twice in March and April of this year in two of the Canadian-built linear accelerators, sold under the name Therac 25.

Two patients were injured, one who died three weeks later, at the East Texas Cancer Center in Tyler, Texas, and another at the Kennestone Regional Oncology Center in Marietta, Ga.

The defect in the machines was a "bug" so subtle, say those familiar with the cases, that although the accident occurred in June 1985, the problem remained a mystery until the third, most serious accident occurred on April 11 of this year.

Late that night, technicians at the Tyler facility discovered the cause of that accident and notified users of the device in other cities.

The US Food and Drug Administration, which regulates medical devices, has not yet completed its investigation. However, sources say that discipline or penalty for the manufacturer is unlikely.

Modern cancer radiation treatment is extremely safe, say cancer specialists. "This is the first time I've ever heard of a death" from a therapeutic rediation accident, said FDA official Edwin Miller. "There have been overtreatments to various degrees, but nothing quite as serious as this that I'm aware of."

Physicians did not at first suspect a rediation overdose because the injuries appeared so soon after treatment and were far more serious than an overexposure would ordinarily have produced.

"It was certainly not like anything any of us have ever seen," said Dr. Kenneth Haile, director of radiation oncology of the Kennestone radiation facility. "We had never seen an overtreatment of that magnitude."

Estimates are that the patients received 17,000 to 25,000 rads to very small body areas. Doses of 1,000 rads can be fatal if delivered to the whole body.

The software fault has since been corrected by the manufacturer, according to FDA and Texas officials, and some of the machines have been retured to service.

... (description of the accidents)

The Therac 25 is designed so that the operator selects either X-ray or electron-beam treatment, as well as a series of other items, by typing on a keyboard and watching a video display screen for verification of the orders.

It was revealed that if an extremely fast-typing operater inadvertently selected the X-ray mode, then used an editing key to correct the command and select the electron mode instead, it was possible for the computer to lag behind the orders. The result was that the device appeared to have made the correct adjustment but in fact had an improper setting so it focussed electrons at full power to a tiny spot on the body.

David Parnas, a programming specialist at Queens University in Kingston, Ontario, said that from a description of the problem, it appeared there were two types of programming errors.

First, he said, the machine should have been programmed to discard "unreasonable" readings - as the injurious setting presumably would have been. Second, said Parnas, there should have been no way for the computer's verifications on the video screen to become unsynchronized from the keyboard commands.

[This story was also reported by Jim Kirby. It is very rare that I get MULTIPLE copies of such a report. Statistically, that suggests that there must be many things that never get reported... PGN]

Risks of VDTs

Alan Wexelblat <wex@mcc.arpa>
Mon, 16 Jun 86 11:50:07 CDT

Excerpted from an article by Loren Stein of the Center for Investigative Reporting in San Francisco, published in the July 1986 issue of the Progressive:

"[...]Effictive with the 1986 budget, the Reagan administration has cut off $1.5 million in funds for the non-ionizing radiation [the kind emitted by VDTs] research program in North Carolina's Research Triangle Park, a program in operation since [...] 1971. `For several years,' says Jerold Mande, an assistant to Albert Gore Jr. of Tennessee, `the administration has tried to eliminate the program and each year the House defended it. But the last time around, they gave up.'

"[...]Until recently, many scientists believed that non-ionizing radiation could not affect the body unless its electric field produced heat or an electric shock. But in 1984, _Spectrum_, a leading engineering journal declared that `a growing mass of evidence has virtually ended that debate.'

"`Evidence of the effects [of non-ionizing radiation] on the nervous system and the immune system of animals was already well-established by the end of the '70s,' wrote Eric Lerner, a former contributing editor of _Spectrum_ `while evidence of effects on the genetic material has accumulated most rapidly over the past few years.' These discoveries may mean that our bodies are far more sensitive to non-ionizing radiation than previously thought [...].

"Two of the EPA's most important experiments in non-ionizing radiation - now shelved - underwent years of detailed preparation and were on the verge of actual testing. One involved the lifelong exposure of rats to low-level radio-frequency radiation. `I really looked forward to this experiment,' says Tell. `We had finally, after five years, gotten all the facilities set up to support the experiment. It took so much time, manpower, and money. Now it's through.'

"Another key project tried to replicate some dramatic findings for Jose Delgado's research laboratory in Madrid, Spain. In 1982, associates at this labe discovered that extremely weak-pulsed magnetic fields - only one five-hundredth the strength of the Earth's natural magnetic field - caused chick embryos to develop malformed hearts and central nervous systems.

"[...]The EPA [...] will not participate in an international effort to verify Delgado's findings - an effort made possible by the EPA's development of equipment that is being shipped to Canada, Sweden, and three other places to create identical test environments. [...]

"Funding for non-ionizing radiation research has been slashed in other Government programs as well. An eagerly anticipated reproductive study involving 4000 VDT operators of child-bearing age by the National Institute of Occupational Safety and Health was among the casualties.

"The EPA research branch is no longer necessary, claim some officials, because the agency will soon publish voluntary guidelines for exposure to RF radiation; overexposure can raise body temperature, which animal research indicates may be harmful to pregnant women and their unborn children. [EPA claims there's no conclusive evidence of harm.]

"Other experts [say] the EPA guidelines will suffer from the dismantling of the agency's non-ionizing radiation research team. [...]Tell's office is issuing the soon-to-be-published RF radiation guidelines; he says `Obviously, we need biological experiments. They've helped us tremendously.'

[Senator Gore has an interest in this and fought to keep the research. Technical comments will be given to the EPA on the guidelines and the EPA will not have the expertise to evaluate them.]

Alan Wexelblat
ARPA: WEX@MCC.ARPA
UUCP: {ihnp4, seismo, harvard, gatech, pyramid}!ut-sally!im4u!milano!wex

---

Minor addition on Risks of Distributed Energy

<TMPLee@DOCKMASTER.ARPA>
Wed, 18 Jun 86 10:54 EDT

Two observations to add to Chuck Ferguson's comment on distributed energy. In the debate over the safety of nuclear energy it has been proposed that a further alternative to the ones mentioned in Risks is solar energy. Those so doing ignore the fact that (whether the weather cooperates or not in terms of percentage of sun) in the part of the country he and I come from it would be necessary to clear the snow off some types of solar energy devices in the winter. The number of likely deaths from people climbing on their roofs to shovel off their solar cells is guaranteed to exceed the probable number of deaths from a nuclear power plant accident.

The point here is not the specific technologies involved, but the two recent messages on the topic just prompted to think of this one more example (its going to be hot and humid here today, and somehow snow came to mind) of how in comparing risks of various potential solutions one must take everything into consideration. (Isn't it also true that coal-fired plants actually release a fair amount of low-level radiation that somehow gets ignored? and how many more deaths and injuries are there amongst coal-miners than uranium miners ... oops, got carried away. Note of course that any of these conjectures may be wrong and one would have to insist on credible statistics before making any conclusions.)

Ted

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 10

Friday, 20 June 1986

Contents

- Re: Privacy Legislation & Cellular Swiss Cheese (RISKS-3.8)
  Geoff Goodfellow
- Re: Privacy Legislation (RISKS-3.6) [divulging]
  Dan Franklin
- Re: Privacy Legislation (RISKS-3.6) [radar detectors]
  Herb Lin
- Info on RISKS (comp.risks)

Re: Privacy Legislation & Cellular Swiss Cheese (RISKS-3.8)

the tty of Geoffrey S. Goodfellow <crcwdc!geoff@seismo.CSS.GOV>
19 Jun 86 11:19:03 EDT (Thu)

I co-authored an article on the ease of which cellular can be spoofed, COMINT'd and SIGINT'd in the November issue of PERSONAL COMMUNICATIONS TECHNOLOGY. An on-line copy of the article may be FTP'd with 'anonymous' login from [SRI-CSL]<Geoff>Article.Celllar-Sieve or by sending me a message requesting one by reply copy.

With respect to the impending facade of Privacy Through Legislation, here is a good report on it which appeared on the Info-Hams mailing list. Pay special attention to such gems as how Cordless phones are not included, and the different level of protection afforded to Cellular abusers vs. the traditional mobile telephone IMTS systems on 150 and 450 Mhz.

Geoff Goodfellow
Cellular Radio Corp.
Vienna, VA

[Note: This HamNet Electronic Edition is a limited excerpt from the full published edition [Vol 8 #11 -- 6/01/86] of The W5YI Report. Selected and prepared by Scott, W3VS. Commercial redistribution of this copy is prohibited.]

Up to the minute news from the worlds of amateur radio, personal computing and emerging electronics. While no guarantee is made, information is from sources we believe to be reliable. May be reproduced providing credit is given to The W5YI Report.

o Electronic Privacy Bill Passes Subcommittee

Legislation extending protection against unwarranted interception of electronic communications by outsiders passed its first and most difficult test during mid-May. RF signals present throughout our homes will no longer be public domain if HR 3378 ultimately becomes law.

After weeks of negotiation, the House Judiciary Subcommittee on Courts, Civil Liberties and the Administration of Justice reached a compromise agreement with the Department of Justice setting the stage for subcommittee approval. The mark-up session was packed with 120 spectators crowded into a room designed for 60.

Most of Justice's problems had to do with adding barriers to law enforcement efforts. The bill, as approved, requires the government to obtain detailed search warrants to intercept and use electronic messages in transit. The subcommittee acknowledged that they still had a couple of things to work out in the "foreign counterintelligence field."

The legislation, the Electronic Communications Privacy Act brings the Wiretap Act of 1968 up-to-date by including such communications services as cellular radio, computer data transfer, electronic mail and satellite communications not in use when the act was first passed. The final draft of HR 3378 was unanimously approved after two suggested amendments (which made sense to us) were defeated. The final subcommittee vote had been delayed three times previously.

The bill is far reaching and will effect nearly every American in one way or another. While legislators, the media, and the various electronic industries are widely portraying the bill as protecting cellular privacy, it doesn't at all. Cellular phones, of course, are the space age version of the old car radio telephone.

The bill particularly affects hobby, industrial and government radio users and listeners in that it details what can- and cannot be monitored. Supporters of the legislation include such industrial giants as IBM, AT&T, MCI, Motorola, GE, GTE, Bell telephone, all three TV networks, ... and various telephone, videotex, electronic mail and computer equipment trade organizations.

Since most of us are concerned with the personal use of electronic communications and the right to monitor the radio spectrum, we will focus on that aspect.

A new definition of electronic eavesdropping has been proposed.

Instead of "acquisition of the content", it is now "interception of the transmission of the content."

A penalty of up to a year in jail and $10,000 fine would be imposed on those intercepting certain transmissions not intended for the general public in the shortwave band...such as remote broadcast pickup stations operating around 26 MHz and perhaps ship-to-shore radio telephone conversations. Any encrypted (scrambled) transmissions are also protected.

Strangely, scanner owners are subject to the year in jail/$10,000 fine if they tune in the old 150/450 MHz carphone service - but only 6 months in jail and a $500 fine if they listen to a 900 MHz celluar phone call!

Specifically exempted from coverage by the bill are all amateur radio, CB and GMRS (General Mobile Radio Service) communications. Ham auto-patch telephone calls therefore are not affected even though a participant expecting privacy might not be aware that the radio portion of the call is being widely transmitted.

The radio portion of a private telephone call terminated by a cordless phone is also not privacy protected "since these calls can be easily intercepted." The subcommittee noted that the FCC requires manufacturers to include privacy disclaimers with cordless equipment.

Actually, just about any radiotelephone call can be easily intercepted, but the legislators perceived some as harder than others. Cellular phone calls can even be received on consumer TV sets.

Broadcast services not intended for the public (such as a piggy-backed FM subcarrier service) may not be monitored.

Radio services not protected by the bill include "any station for the use of the general public, or that relates to ships, aircraft, vehicles, or persons in distress" as well as "any marine, aeronautical, governmental, law enforcement, civil defense, or public safety communications ...readily accessible [not encrypted] to the general public." Thus, you can listen to ongoing law enforcement manuevers ....even Air Force One, but not a random phone call you might hit upon in the spectrum.

What can be monitored by satellite dish owners was specifically not resolved since this question is currently before the House Telecommunications Subcommittee.

Private fixed microwave links, FM subcarriers, and broadcast auxiliary or remote pickup stations were specifically protected.

Rep. Mike DeWine (R-Ohio) offered two amendments at the subcommittee mark-up session dealing with cellular phone calls.

DeWine, a former prosecuting attorney, said that while he was in basic agreement with the intent of the bill, he was troubled by the fact that old television sets still being sold can inadvertantly overhear a cellular phone call. He also said that scanner marketing was not covered by the bill... "If a scanner stops at a cellular phone channel ...this bill means that (a scanner listener) could be imprisoned for six months ...even if he did not disclose the information.

He acknowledged that the Justice Department told them that they wouldn't enforce scanner (or TV) cellular listening but "it's basically bad public policy to create a law that everyone knows will not be enforced... It brings about a disrespect for the law. ...It weakens anybody's faith in the criminal justice system. We are not talking about difficult enforcement. What we are talking about is an impossibility, unless we are willing to violate people's Consititutional Rights and go into their own homes..."

The bill "...creates the illusion of protection," DeWine testified. "The facts are that it will no more protect (cellular) the day after we pass this bill than the day before..."

Rep. DeWine suggested an amendment that would outlaw the overriding of an encrupted telephone conversation. He said laws already exist that prohibit divulging intercepted information. He is concerned that "...the cellular phone industry will use this bill to tell people that they have an expectation of privacy when, in fact, they do not."

Chairman Kastenmeier agreed that the bill could not easily be enforced, but that encruption cost was prohibitive ($2,500 for a mobile, $164,000 for a base station.) Declaring that he didn't want to make America an encrypted society, he urged defeat of the amendment.

Holding up a scanner advertisement which promoted listening to "radiotelephone conversations that offer more real-life intrigue that most soap operas," Kastenmeier said "we cannot encourage this! We have set down the rules of the road whereby that is off limits... Scanners are very useful devices, and they will continue to be, excepting there ought to be some things that are protected against even, yes, even against scanners." A voice vote defeated the amendment by a clear majority.

A second amendment was introduced by DeWine, eliminating the 6 month prision sentence from the cellular penalty. That too was rejected.

With no further amendment being offered, the substitute draft of HR 3378 was unanimously adopted by voice vote. The Subcommittee agreed to report the bill out to the Judiciary Committee for further action.

HR 3378 is still very far from being a law. It must be approved by the Judiciary Committee and the full House ...then reconciled with a similar bill pending before the Senate Copyright Committee. It gets signed into law by the president. The reality of the matter is, however, that government control over radio wave reception in your home will indeed be eventually enacted in some form.

Re: RISKS-3.8

Dan Franklin <dan@bbn-prophet.arpa>
Fri, 20 Jun 86 14:38:35 EDT

> Does anyone have any idea how the last part (radio telephones) could be
> legally supported in view of other legal freedoms? I thought that one
> was free to listen to any frequency one wished in the US (Canada too).
> You don't have to trespass to receive radio signals.

Receive them, yes; tell anyone else what you heard, no. As I understand the law, if a radio signal is part of a conversation--that is, clearly directed at some specific other person--you are forbidden to divulge the contents of that signal to a third party. You might be forbidden to make any other use of it, too; I don't remember for certain.

So eavesdropping is already suspect in current law, and it would not be such a big change to say, for instance, that you could not *intentionally* receive radiotelephone signals. If your neighbor's radiotelephone happened to come in on your stereo, you wouldn't then be breaking the law. I do not actually know what the new law says, but there do exist ways to safeguard privacy without compromising the "right to receive".

Dan Franklin

Privacy legislation (RISKS-3.6)

<LIN@XX.LCS.MIT.EDU>
Tue, 17 Jun 1986 00:32 EDT

[On the same topic...]

Not true. States routinely ban the use of radar detectors, and that is nothing more than "listening to a frequency."

[Well, things seem to be changing. In California, PASSIVE detectors are now legal, and can be bought at Radio Shack among others. Mail order outfits are also doing a boom business. I presume this is true in other states as well. ACTIVE JAMMERS are of course still illegal. [[This messasge does not constitute an endorsement on the advisability of using a detector, or of the reliability of any such product. I won't even contemplate the risks involved of using one.]] PGN]

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 8

Sunday, 15 June 1986

Contents

- Challenger, SDI, and management risks
  Dick Dunn
- Re: Risks from inappropriate scale of energy technologies
  Chuck Ferguson
- Distributed versus centralized computer systems
  Peter G. Neumann
- Privacy legislation
  Michael Wagner
- Info on RISKS (comp.risks)

Challenger, SDI, and management risks

Dick Dunn <nbires!rcd@ucbvax.Berkeley.EDU>
Fri, 13 Jun 86 12:24:29 mdt

The Challenger failure has an implication for SDI that I've not seen discussed much. I regard the solid-rocket failure as primarily a management failure and only secondarily an engineering failure. Why? Because according to the Rogers group reports, there had been serious concern with the possibility of failure of the O-ring seals, but it got lost or suppressed along the way. Challenger's ill-fated launch was apparently made in spite of best engineering advice to the contrary.

How does this apply to SDI? I'll give a sketch; I hope that other people will add more. SDI is under fire from several places (substantial part of Congress, various public-interest groups, many influential technical people). It is therefore important for the supporters (willing and/or appointed) of SDI to present a convincing case that SDI can "do the job." There is tremendous pressure to justify SDI. Translate this into "there is tremendous pressure to argue the case that SDI can be built and can work--whether or not it really can." To be blunt, there is a tremendous incentive to cover up any potential inability to build an SDI system or any inadequacy once it is built. Of course, if the SDI system is built, and is used, and fails, there will be much more lost than seven lives and some megabucks of hardware. (I have a hard time typing the word "terabucks":-) There probably Wouldn't even be a presidentially-appointed blue-ribbon investigative committee...

The hard questions: Do we have a way to manage a project of the magnitude of SDI that will give us any halfway-reasonable assurance that the project will work? Is there any technique that can be applied to reward those who discover problems and punish those who cover them up, instead of the other way around?

(My own experience, unfortunately, tells me that these aren't really hard questions. Rather, they are questions which are easily answered "no!" The difficulty in managing any large project, particularly one which involves a lot of software, is legendary.)

In summary, I'm saying that Challenger failed not for technical reasons-- I believe that the technical problems are real but surmountable--but for managerial reasons. Further, I think that we need to talk about SDI feasibility in more than technical terms; we need to address whether we could manage the project even if all of the technical problems were surmountable. The answer is anything but a clear "yes".

Dick Dunn

> [From The New York Times, Sunday, 15 June 1986:
>
> New York - The ''Star Wars'' anti-missile plan has been seriously and extensively damaged by the Challenger disaster and other setbacks in the American space program, aerospace analysts say. Officials of the anti-missile defense program, formally called the Strategic Defense Initiative, deny any serious damage to the program, but aerospace experts say the problems within the space program have sent shock waves through research programs. ...]

Re: Risks from inappropriate scale of energy technologies

Chuck Ferguson - SCTC <CTFerguson@HI-MULTICS.ARPA>
Thu, 12 Jun 86 20:06 CDT

In RISKS-3.6, Michael Natkin states:

> The public has long been duped into the idea that centralized energy management has its best interest in mind as we develop ever increasing electrical capacity. But centralized reactors and other "hard" technologies are extremely susceptible to terrorist attack and other failures, as has been mentioned before.

Centralized power supplies may be "extremely susceptible" to terrorists but their susceptibility to failure is not as high as being claimed. It is true that the consequences of a failure might be great; however, for a large centralized power plant it is economical to expend greater resources to prevent their failure (e.g., redundancy) than for the components of a distributed system. Furthermore, I submit that all current power systems have some degree of distributed or redundant functionality to allow periodic maintenance shutdowns if for no other reason.

I further submit that there is a significant risk associated with distributed systems which is being ignored. Many such systems are themselves dangerous when poorly maintained or operated improperly. There are also hazards associated with storing combustible fuels near a dwelling or other populated area. Witness the following:

- o How many chimney fires have you heard about since the "energy crisis" began? A fireplace is a relatively low-tech device yet some people manage to make them dangerous.

- o Why is it that several houses burn down at the start of every cold season? An oil-fired furnace is a relatively low-tech item also, yet every year someone's gets choked with soot and catches fire.

- o Ever heard of a methane gas explosion in a sewer system? I recently heard an amusing story about a manure fire at a horse ranch - ten years worth of horse manure had been piled in one place until one day it spontaneously caught fire.

One would be surprised how much damage some people can do with low-tech alternative energy. To paraphrase one of the better known 'computer security experts' [emphasis added], "Terrorists can never compete with incompetents". I wonder whether more people lose their lives each year in the commercial production of power or in incidents similar to the above.

With respect to the public "being duped" - sounds like another conspiracy theory to me (yawn).

Chuck Ferguson, Honeywell, Inc., Secure Computing Technology Center

---

Distributed versus centralized computer systems

Peter G. Neumann <Neumann@SRI-CSL.ARPA>
Sun 15 Jun 86 22:32:04-PDT

Although Chuck's note does not seem as closely related to RISKS as some of his past contributions, it suggests various additional comments. A myth of distributed computing systems is that distribution avoids centralized vulnerabilities. WRONG! The 1980 ARPANET collapse gave us an example of an accidentally propagated virus that contaminated the entire network. The first shuttle synchronization problem is a further example.

By distributing what has to be trusted, there may be more vulnerabilities -- unless the distributed components are TOTALLY autonomous -- in which case we are not really talking about DISTRIBUTED systems, but rather SEPARATE systems. Security flaws in the systems and networks can result in transitive vulnerabilities, or permit global compromises by iteration.

Further, the point raised by Chuck regarding maintenance is an important one in distributed computer systems, especially if some of the distributed sites are remote. Well, then, you say, let the field engineers dial up the remote site. But then that path provides a monstrous new vulnerability. Then we get solutions like the remote backup scheme proposed a while back that gets special privileges... Also, remember the fundamental flaws in the standard two-phase commit protocols, three-clock algorithms, and so on. Once again it might be useful to consider truly robust algorithms such as interactive consistency and Byzantine agreement. However, for every more complex would-be technical solution there are often further technical problems introduced. For every assumption that things have gotten better there seem to be even grosser counterexamples and further vulnerabilities outside of the computer systems. Thus,

It is folly to trust software and hardware if an end-run can bypass or compromise the trusted components. But it is also folly to assume that sabotage is significantly less dangerous just because a system is distributed. That may be true in certain cases, but not generally.

Peter [Please excuse me if I have repeated some things that I said in earlier RISKS in a different context.]

Privacy legislation (RISKS-3.6)

Michael Wagner <ubc-vision!utcs!wagner@seismo.CSS.GOV>
Sat, 14 Jun 86 11:26:37 edt

>A news clipping from this morning's "Los Angeles Times" (page 2, The News
>in Brief):
>
> The House Judiciary Committee voted 34 to 0 for a bill seeking to
> bring constitutional guarantees of the right to privacy into the
> electronic age. The legislation would extend laws that now protect
> the privacy of the mails and land-line telephone conversations to also
> cover electronic mail and some telephones that use radio waves.

Does anyone have any idea how the last part (radio telephones) could be legally supported in view of other legal freedoms? I thought that one was free to listen to any frequency one wished in the US (Canada too). You don't have to trespass to receive radio signals.

Contrast this with the mails. The privacy of the mails is supported by property laws. That is, you put your mail into a box which belongs to the post office. If anyone breaks into that box (or the van which picks up the mail, etc) they are breaking property laws. Similarly for land lines. One has to 'trespass' to tap a land line.

It seems to me that the legislators have 'extended' the laws over an abyss. Or have I missed something?

The relevancy to RISKS, of course, is that most people don't think about the technology that radio-telephones use. I'm sure most people assume "it's a phone - it's (relatively) safe". Not true, of course. In fact, some people have used their own handsets to make phone calls on other peoples phones!

Michael Wagner

[I do not recall having pointed out in this forum the ease with which the cellular phone schemes can be spoofed, e.g., getting someone else to pay for your calls. There is another security/integrity problem waiting to be exploited. PGN]

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 6

Thursday 12 June 1986

Contents

- Risks from inappropriate scale of energy technologies
  Michael J. Natkin
- Shuttle Software
  David C. Smith
- An additional SDI problem: sensor technology
  Eugene Miya
- Privacy in the electronic age
  Dave Platt
- Sgt York software
  Larry Campbell
  Mark Vilain
- Info on RISKS (comp.risks)

Risks from inappropriate scale of energy technologies

"Michael J. Natkin" <mjn%brown.csnet@CSNET-RELAY.ARPA>
10 Jun 86 (Tue) 23:46:50 EDT

One of the most important categories of long term risks to the public from technology seems to have been overlooked in Risks so far. The assumption that more technology is automatically good is so ingrained in our thinking that it is hardly questioned. We measure our welfare in terms of Gross National Product, not by how many people have enough to eat, or by distribution of income.

In particular a vast amount of our technical, capital and human resources are expended developing monolithic energy technologies without regard to end use needs. The public has long been duped into the idea that centralized energy management has it's best interest in mind as we develop ever increasing electrical capacity. But centralized reactors and other "hard" technologies are extremely susceptible to terrorist attack and other failures, as has been mentioned before.

The public has been told that it doesn't have the expertise to make decisions about such high risk high technologies as SDI and nuclear power, and in some sense this is true. But the technocrats have preempted the public's right to make the moral and political policy which guides the choices.

I think that we should be pursuing a policy course which develops technology that can be put safely in the hands of non-technical people. This might take the form of small burners which use the methanol from organic wastes, windmills, or non-electrical solar collectors, to name a few possibilities. Localized, distributed technologies have many advantages, including ease of repair, localization of risk from outage, and major reductions in distribution losses and cost of distribution equipment and labor. I strongly recommend Amory Lovins' "Soft-Energy Paths" to others interested in issues of appropriate scale in technology.

Michael Natkin
CSnet: mjn@brown
ARPA: mjn%brown@csnet-relay
UUCP: ...!{allegra,decvax,ihnp4}!brunix!mjn

Shuttle Software

David C. Smith <DCSmith@SRI-AI.ARPA>
Wed 11 Jun 86 08:55:30-PDT

The cover story of the September, 1984, CACM is "A Case Study: The Space Shuttle Software System". As with other CACM case studies, this one is a discussion, or interview, with several people involved with the subject matter, in this case 6 individuals from the IBM Federal Systems Division. An Outline of the Interview included in the article contains:

Project Overview
The Shuttle Computers
Project Organization
Testing Facilities
Detailed System Operation--No Redundancy
Redundant Set Operation
System Problems
The Interprocess Variable Problem
Concluding Remarks

The issue also contains several other articles in a Special Section on Computing in Space, including "Design, Development, Integration: Space Shuttle Primary Flight Software System", written by 2 senior technicians from the IBM FSD.

It seems like a good place for a novice to the shuttle and its systems (like myself) to get some basic information about the shuttle computers and the complexity of the systems.

Dave Smith

An additional SDI problem: sensor technology

Eugene Miya <eugene@ames-aurora.arpa>
11 Jun 1986 1124-PDT (Wednesday)

The view expressed within are the view of the author and not of my agency nor of the Federal government. ------------------------------ A lot of interest has been expressed regarding the focus of the problems of SDI: the software, in particular battle management. Note the Science article of May 9 1986. However, I wonder about the other components of the system. Where there are various groups watchdogging computing, but the more hardware oriented, EE areas such as radar have fewer opposition elements. Recent postings on cruise missiles and the integration of DIVAD move me to post this.

Sensor technology is one area which worries me. SDI battle management makes certain assumptions about the ability to detect and identify targets. I think that most computer people don't understand the nature of radar to worry about the problems of `target' detection and ranging. That is all that radar is: detection (boolean) and ranging (distance=rate times time). A first starting references is Skolnick's text on Radar. (Dated)

Inherent problems with a ranging system include: Range and azimuth ambiguities, difficulties with empirically determined signatures. Most people don't seem to understand that knowing the geometry of systems are important. Satellite images [some radar maps to be used in offensive missiles] are not photographs (you must call them images) because their geometry is from a linear and not a point perspective, so distance determination for things like cruise missiles cannot be done using a straight edge. Radar (simple) is like looking at the world using a monochromatic spot light from the point where you are looking: you don't get shadows (an important distance cue). Note: I have not talked about clutter, or noise (ever wonder how high speed jets detect jets from ground objects, or how AWACS which points down get insignificant ground objects cleared?).

While there exist solutions, all of them involve tradeoffs in complexity, cost, and new emergent problems. Solutions in Doppler systems, phased arrays, stereo transmit/receive systems, but just the inherent simplicity of the concept and the over-generalization of use worries me. This is a case where "high-level language" solutions may not be high-enough.

--eugene miya, NASA Ames Research Center, eugene@ames-aurora.ARPA
{hplabs,hao,dual,ihnp4,decwrl,allegra,tektronix,menlo70}!ames!aurora!eugene

Privacy in the electronic age

Dave Platt <Dave-Platt%LADC@HI-MULTICS.ARPA>
Wed, 11 Jun 86 10:47 PDT

A news clipping from this morning's "Los Angeles Times" (page 2, The News in Brief):

  The House Judiciary Committee voted 34 to 0 for a bill seeking to
  bring constitutional guarantees of the right to privacy into the
  electronic age. The legislation would extend laws that now protect
  the privacy of the mails and land-line telephone conversations to also
  cover electronic mail and some telephones that use radio waves.
  The bill was cleared at the request of Rep. Robert W. Kastenmeier
  (D-Wis.), chairman of Judiciary's subcommittee on courts, civil
  liberties and administration of justice.

Anyone know the details? Just what privacy coverage would be afforded by this bill in its present form? How would the bill's provisions affect the sysops of private electronic bulletin-board systems, for example? Would this bill clarify the legal standing of electronic transactions and messages re their use as evidence in court?

[Very strange. RISKS-3.1 noted that the House sent a bill to the Senate on 3 June that covered "federal interest" computers. Is this an additional bill, or a modification of one already sent over? Maybe someone in the House is reading RISKS and noted the apparent flaws in the bill that I mentioned in RISKS-3.1? PGN]

Sgt York software

<decvax!bellcore!genrad!panda!wjh12!maynard!campbell@ucbvax.berkeley.edu>
Wed, 11 Jun 86 01:52:39 edt

In RISKS 3.4, Mike McLaughlin (mikemcl@nrl-csr) and Ken Laws (laws@sri-ai) dispute the Sargent York latrine fan story. [...]

I quote from a story by Gregg Easterbrook in the November 1984 issue of _The Washington Monthly_:

> During a test one DIVAD locked on to a latrine fan. Michael Duffy, a report for the industry publication _Defense Week_, who broke this aspect of the story, received a conference call in which Ford officials asked him to describe the target as a "building fan" or "exhaust fan" instead.

The Washington Monthly and _Defense Week_ are both reputable publications. Does anyone have a citation for a retraction in _Defense Week_, or should we assume that the TV networks swallowed Ford's story whole?

Larry Campbell   The Boston Software Works, Inc.
ARPA: campbell%maynard.uucp@harvard.ARPA   120 Fulton Street, Boston MA 02109
UUCP: {alliant,wjh12}!maynard!campbell   (617) 367-6846

Sgt. York software

Marc Vilain <MVILAIN@G.BBN.COM>

Wed 11 Jun 86 12:48:29-EDT

Here is some information on the DIVAD software that hasn't appeared yet in this forum. [It] is abstracted from a longer note compiled by Reid Simmons from material he received from Gregg Easterbrook (both his article in the Atlantic, and personal communications).

According to Easterbrook, the DIVAD did target a latrine exhaust fan in one series of tests. The target was displayed to the gunners that man the DIVAD. But the Sgt. York did not shoot at the latrine, or even swivel its turret in the latrine's direction, having prioritized the target as less important than other targets in its range.

In another series of tests (Feb. 4 1984), U.S. and British officials were to review the DIVAD as it took upon a rather cooperative target: a stationary drone helicopter. On the first test run, the DIVAD swiveled its turret towards the reviewing stand as "brass flashed" and the officials ducked for cover. It was stopped only because an interlock was put in place the night before to prevent the turret from being able to point at the reviewing grandstand. Afterwards, the DIVAD shot in the general direction of the helicopter but the shells traveled only 300 yards. The official explanation is that the DIVAD had been washed the night before, screwing up its electronics. Easterbrook wonders what would happen if it rained in Europe when the DIVAD was being used.

Easterbrook goes on to claim that the snafus the DIVAD experienced were very much due to software. The main problem was that the pulse-Doppler tracking radar and target acquisition computer were a very poor match. Easterbrook claims that the hard problem for the software (tracking fast, maneuvering planes) was easiest for the pulse-Doppler radar which needs a moving target. On the other hand, the hard part for the radar (detecting stationary helicopters) was the easiest to aim at. The DIVAD mixed two opposing missions.

Easterbrook goes on to say that human gunners are often more successful than their automated counterparts. They can pick up on visual cues, such as flap position on approaching aircraft, to determine what evasive maneuvers the enemy might make. These kinds of cues are not visible to things like pulse-Doppler radars. Further, evasive courses of action are hard for human gunners to counter, but even harder for target tracking algorithms (again the lack of visual cues comes as a disadvantage). For example, the DIVAD expected its targets to fly in a straight line (which my military friends tell me is not too likely in a real combat).

There is lots more to the Sgt. York story, not all of which is relevant here. If there is a moral to be drawn specifically for RISKS, it's that as advanced as our technology may be, it may not always be the match of the problems to which it is applied. This was certainly the case with the unfortunate DIVAD.

marc vilain

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 11

Monday, 23 June 1986

Contents

- A medical risk of computers (overdose during radiation therapy)
  Jon Jacky
- Secure computer systems
  Herb Lin
- Radar Detectors (Re: Privacy legislation in RISKS-3.10)
  Jeff Makey
- Telco Central office woes in Southfield, MI.
  via Geoff Goodfellow
- Reducing the managerial risks in SDI
  Bob Estell
- Economic Impact of SDI: Transcript Info
  Richard A. Cowan
- Info on RISKS (comp.risks)

Re: A medical risk of computers (overdose during radiation therapy)

Jon Jacky <jon@uw-june.arpa>
Sat, 21 Jun 86 13:11:44 PDT

> (Karen Sollins quotes story from Boston Globe - to paraphrase, patient
> would be badly overdosed if operator first selected electron beams then
> changed selection to X-rays. David Parnas observed that two kinds of errors
> were made; first, system should not have accepted inconsistent or unsafe
> input specifications, second, synchronization problem elicited when operator
> types rapidly.

I work in a radiation therapy department, so my observations may be of interest.

First, this is a VERY SCARY STORY. It was estimated that patients got 17,000 to 25,000 rads in a single treatment. For comparison, typical therapeutic doses are in the range 4000 - 6000 rads, delivered in 20 to 30 separate daily treatments administered over a month or more. What is really alarming here is that the therapy machines are set up to deliver dose rates on the order of 100 rads per minute. I believe that most therapists would assert that there was no way, physically, that a machine could deliver tens of thousands of rads in a few seconds. That was my reaction when I first read the story in the New York Times (Sat. June 21, p.8, national edition). The New York Times story mentioned that when the accidents first occured, the operators thought the patients had somehow been electrically shocked (by leakage currents through the couch or something) rather than overdosed.

The New York Times story did not mention the x-ray/electron confusion, and that is the key to this accident. A modern radiation therapy machine is based on a linear accelerator that produces an electron beam with an energy of 25 MeV or so. You may direct the electrons directly into the patient (at this energy electrons are ionizing radiation), or, to produce X-rays, you put a heavy metal target in the electron beam, and when the electrons hit the target X-rays come out the other side. The target is moved in and out of the beam automatically. Here is my speculation of what happened: I suspect that the current in the electron beam is probably much greater in X-ray mode (because you want similar dose rates in both modes, and the production of X-rays is more indirect). So when you select X-rays, I'll bet the target drops into place and the beam current is boosted. I suspect in this case, the current was boosted before the target could move into position, and a very high current electron beam went into the patient.

How could this be allowed to happen? My guess is that the software people would not have considered it necessary to guard against this failure mode. Machine designers have traditionally used electromechanical interlocks to ensure safety. Computer control of therapy machines is a fairly recent development and is layered on top of, rather than substituting for, the old electromechanical mechanisms. I suspect there was supposed to be an interlock between beam current and target position, which should have prevented the beam from going on at all. Maybe there was, but it was broken, too.

I stress that I am not familiar with the design of this particular machine and that these are just speculations.

I should also mention that these are the first incidents I have heard of where an overdose was delivered due to an error in the therapy machine dose rate. Overdoses in radiation therapy do occur, but in all the cases I have heard of they are due to incorrect planning and patient positioning: that is, the radiation beams pass through the wrong part of the patient and irradiate healthy tissues rather than the tumor, or the therapists incorrectly estimate the dose rate inside the body that will be produced by a specified machine dose rate.

-Jonathan Jacky
Department of Radiation Oncology
University of Washington, Seattle WA

Secure computer systems

<LIN@XX.LCS.MIT.EDU>

Tue, 17 Jun 1986 00:22 EDT

I have a question for the RISKS readership.

I want to make an arrangement in which I can feed data to a computer in the physical possession of an adversary. The output of the program can be certified via a public-key encryption system. The question if this: can the computer hardware be designed so that its programming cannot be compromised, even though the data would be entered by the adversary? Alternatively, can the computer detect attempts to compromise it?

(Assume that the data is known to be good.)

[Herb, You have almost gotten to the MUTUAL SUSPICION problem, where a vendor provides the program and a customer provides the data -- and where neither trusts the other. Limited solutions can be conceived, but many assumptions must be made about the integrity of the communication paths, the trustworthiness of the environment in which the mutually trusted arbiter must run, the absence of all sorts of side effects (such as Trojan horses) and covert channels, the adequacy of the hardware if a general solution is sought, the nontamperability of the hardware and the trusted software, and so on. In your specific case, the answer is to a first approximation NO, although if you start making (unreasonable?) assumptions, MAYBE. Peter]

<<I'm not concerned about the hardware being maintainable (though it can be replaceable at great cost). Herb<>

Radar Detectors (Re: Privacy legislation in RISKS-3.10)

Jeff Makey <Makey@LOGICON.ARPA>
21 Jun 86 20:49 PDT

Radar detectors are presently legal in 48 states. Only in Connecticut, Virginia, and (I think) the District of Columbia are they illegal. As I understand it, Virginia's law is based on the idea that it is illegal to use radio frequencies in the commission of a crime. Thus, it would seem that using a radar detector in Virginia is illegal only if you are committing a crime (e.g., speeding) when the police use radar on you. This sounds too good to be true, so it probably is :-). I know nothing about the specifics of Connecticut's or DC's laws on radar detectors.

If you are interested in the risks of NOT using a radar detector I would be happy to explain why I am a very satisfied owner of one. This issue isn't really appropriate for RISKS (even though the good ones *do* contain computers!) so let's keep this sort of discussion private.

:: Jeff Makey
Makey@LOGICON.ARPA

Telco Central office woes in Southfield, MI.

the tty of Geoffrey S. Goodfellow <crcwdc!geoff@seismo.CSS.GOV>
22 Jun 86 09:47:20 EDT (Sun)

Clipped from the Telecom digest...

------- Forwarded Message

Date: 21 Jun 86 03:22-EDT
From: Moderator <seismo!XX.LCS.MIT.EDU!Telecom-REQUEST>
Subject: TELECOM Digest V5 #122

TELECOM Digest Saturday, June 21, 1986 3:22AM
Volume 5, Issue 122

Date: Fri, 13 Jun 1986 06:06 MDT
From: Keith Petersen <W8SDZ@SIMTEL20.ARPA>
Subject: Northern Telecom DMS-100 digital switch problems

On Wednesday, May 28, the Southfield, MI (suburb of Detroit) Michigan Bell ESS office's Northern Telecom DMS100 digital switch went down for almost the whole afternoon, reportedly depriving 35,000 subscribers of service (they couldn't even get a dial tone).

Thursday, May 29, it occurred again sometime in mid-morning and the digital switch was down for almost the entire business day (it came back around 5:30 pm local time), this time reportedly taking out 50,000 subscribers, including the police and fire departments.

In an interview, a spokesman for Michigan Bell was quoted as saying they don't know what caused the problem. He went on to say they are working closely with Northern Telecom to find the cause.

A spokesman for Northern Telecom, in a recent telephone conversation, said that some 20-30 software updates for the DMS100 were necessary to cure certain problems with passing 212a and V22.bis modem signals through the switch. It is unclear at this time if these updates have any bearing on the outages of the past two days. According to sources at Michigan Bell and Northern Telecom, the updates have not been done to the DMS100 digital switch in the Southfield central office. They are reportedly scheduled to be done on June 7th.

Stay tuned...

- --Keith Petersen
Arpa: W8SDZ@SIMTEL20.ARPA
uucp: {ihnp4,allegra,cmcl2,dual,decvax,mcnc,mcvax,vax135}!seismo!w8sdz

---

<143C::ESTELL 16-JUN-1986 09:07>

I offered some thoughts to RISKS which were reprinted in ARMS-D. I have gotten some interesting feedback to those thoughts, which I would share. First, let me thank you one and all for the character of your replies; they have been cogent, courteous, and convincing. No hints whatsoever about doubts of my intelligence or integrity - even by those adamantly opposed to my point of view.

Let me summarize (and restate) my principle points:
1. SDI will roll on, at least until '89; i.e., the Reagan Admin. is firmly committed to it. "Nature abhors a vacuum." Americans demand adequate defense, while complaining of its cost [which is usually excessive]. Most groups [e.g., Common Cause] who have tried to stop MX et al have offered no alternative; by default, that leaves us stockpiling weapons; we already know that doesn't work; for it costs too much, raises the balance of terror, and besides the USSR is getting ahead of us now.
2. You and I don't have the wherewithal to stop SDI; but perhaps we can glean some benefits from it, especially if we work within the system; e.g., to pursue compatible overall goals, BUT doing valuable things.
3. Bringing our traditional ["non SDI?"] defenses up to reasonable state-of-technology is probably a good idea; e.g., using computers that encourage good software practices, run efficiently, etc.
4. SDI does NOT equate to "ICBM defense."
   You will search my earlier messages in vain for the term "ICBM."
   I made it plain - or tried to - that ICBM's from the USSR [or wherever] are [in MY opinion] less of a threat than less exotic weapons in the hands of criminals/terrorists, of whatever race, religion, nationality.

Now to add some new points:
5. SDI need not cost as much as some fear it might. For example, going to the moon in the '60's cost the USA nothing!
   Miniaturization of electronics, and encapsulation for space led directly to domestic products like the now common "pacemaker."
   The DIFFERENCE between tax dollars paid by those wearing pacemakers, and the "aid to their families" that would have been paid had those heart patients died or been disabled, is more than $25 billion.
   [Data from a CPA friend of mine.]
6. An adequate defense MUST be one that we can afford; and I don't mean by ignoring the deficit, and spending billions just because that's do-able.
   Example: Why are we dismantling Trident subs, while still more funds go to "MX?" Trident IS MX - demonstrated, workable, paid for. If a particular sub becomes obsolete [like some old computers I mentioned], then replace it; but what's the need for "mobile silos on rails?"
   Common sense tells me that there IS a good reason; security regs probably tell WHY I don't know that reason; but Murphy's Law suggest that maybe, just maybe, it's the "military-industrial complex" going after profit.
   That's NOT necessarily bad; it's "free enterprise." But that choice is not necessarily optimum, either. That's why our debate is valuable.
7. Advances in computer technology made in pursuit of SDI can be applied to other problems; e.g., crime prevention. I'm arguing that reliable real-time networks, intelligent "signature recognition" systems, and other digital "tools" can help us intercept dope traffic, as well as ICBM's.
8. Last, but certainly not least, if this work is to be done, it can either be done by the "best and the brightest" or by technocrats and bureaucrats in government, industry, and academia. If that happens, if the best do

not rise to the challenge, then I guarantee that the costs will be much
higher than necessary, and the results much lower than deisrable.
But if we do take the opportunity, then we can use the managers' short
term interests to an advantage; i.e., we can honestly say that "Star
Wars" [R2D2 et al] is not possible today; and then diligently work to
produce what is reasonable. Many managers [in government and elsewhere]
will go along with that incremental progress, because it IS a "bird in
the hand." Indeed, Mr. Reagan is reputed to lead by concept rather than
in detail; so let's supply him the details, rather than abandon that
task to the technocrats - of whatever stripe.
This argument is all the more relevant in light of recent observations
that Challenger failed for managerial reasons, not [just] technical ones.
If the best managers neglect SDI to bureaucrats, then decisions will be
less than optimum; if the best scientists neglect SDI to technocrats,
then even the best decision makers will be hamstrung by second-rate sys-
tems. Our only hope is to marshall our best minds, then evolve SDI.

Finally, to state a position. Some readers have [tried to] guess which side of SDI I'm on; most have been wrong. That's because I won't take a side, as the question is presently posed; viz., am I for or against the President's SDI program? That's too close to "have I stopped beating my wife?" A complex question defies a simple answer. I'm FOR adequate, affordable, ethical defense; I don't believe that SDI, as presented in the popular press, is THE answer. Unlike some readers, I have no direct source of information about what Mr. Reagan and Mr. Weinberger REALLY think; I only have the press summary of their summary of closed sessions in the Pentagon and White House. That's third-hand information. And, I assert again, we must begin with a land-based system; that minimizes the costs, reduces the technical risks, and causes the least threat because such a system could not be used offensively.

Bob Estell
p.s. The opinions above are not necessarily shared by any other person or any organization, real or imaginary.

Economic Impact of SDI: Transcript Info

Richard A. Cowan <COWAN@XX.LCS.MIT.EDU>
Tue 17 Jun 86 19:47:52-EDT

About 5 months ago I advertised a transcript/tape for a debate on the economic implications of Star Wars, held at MIT on November 21, 1985.

Finally, I have uploaded it from my Mac, and it is available online.
The debate is between:

Lester Thurow, MIT Economist
Leo Steg, GE Space Systems Division (retired)
Bernard O'Keefe, Chairman of EG&G

For FTP'ing it, it is located in MIT-XX:<cowan>economics.sdi

If you can't FTP it, tell me and I'll send it to you.
-rich

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 12

Tuesday, 24 June 1986

Contents

- License Plate Risks
  - Chuck Price
- SDI is for ICBMs, Not Terrorists
  - Mark Day
- Still another kind of clock problem
  - Rodney Hoffman
- Estimating Unreported Incidents
  - Ken Laws
- Estimating Unreported Incidents -- and the risks of using statistics
  - PGN
- Re: Privacy legislation (RISKS-3.8) and radio eavesdropping
  - Jerry Mungle
  - Jeff Mogul
  - Jim Aspnes
- Info on RISKS (comp.risks)

License Plate Risks

Chuck Price <price@src.DEC.COM>
Mon, 23 Jun 86 09:56:05 pdt

I heard the following tale on KCBS this morning. [I intersperse a few details from the SF Chron, 23 Jun 86. PGN]

It seems that this fellow [Robert Barbour] desired personalized license plates for his car. Since he loved sailing, he applied for ``SAILING'' and ``BOATING'' as his first two choices [seven years ago]. He couldn't think of a third name of NAUTICAL intent, so he wrote ``NO PLATE'' in as his third choice.

You guessed it. He got ``NO PLATE''.

A week or so later, he received his first parking ticket in the mail. This was followed by more and more tickets, from all over the state [2500 in all!]. It seems that when a police officer writes a parking ticket for a car with no license plates, he writes ``NO PLATE'' on the ticket.

Our friend took his problem to the DMV, which informed him that he should change his plates.

The DMV also changed their procedures. They now instruct officers to write the word ``NONE'' on the unplated parking tickets.

Wonder who's gonna get those tickets now?

-chuck price

[Obviously some poor sap whose license plate says ``NONE''!]

SDI is for ICBMs, Not Terrorists

Mark S. Day <MDAY@XX.LCS.MIT.EDU>
Mon 23 Jun 86 12:04:46-EDT

Bob Estell states that "SDI does not equate to ICBM defense."

This is simply not true. Even in Reagan's first speech about rendering nuclear weapons "impotent and obsolete" (Mar 23, 1983), he went on to say that he was

"directing a long-term research and development program to begin to achieve our ultimate goal of eliminating the threat posed by STRATEGIC NUCLEAR MISSILES." [Emphasis added]

From its inception, SDI has been intended to defend against and deter a massive attack by ICBMs. As others have previously pointed out in RISKS, terrorists don't need to deal with ICBMs and would be foolish to try. At the Stanford debate on SDI feasibility, Maj. Pete Worden (special asst. to the Director of SDIO) answered a question about terrorists and smuggling bombs into the country by saying "We are trying to deter something that is reasonably military, not a terrorist act."

SDI is intended as a defense against Soviet ICBMs and (on particularly optimistic days at SDIO) Soviet cruise missiles. It is not intended to save the United States population from every nuclear threat.

--Mark

Still another kind of clock problem

<Hoffman.es@Xerox.COM>
23 Jun 86 10:00:39 PDT (Monday)

You might be amused by the anomalous dates [in an earlier message from Rodney to me, not included]. Our power was off all weekend for some work. When I came in this morning, no computer servers were working yet --

including the time servers. So I set the date and time on my machine myself, including stuff like "Hours offset from Greenwich Mean Time" and "First day of Daylight Savings Time"! (Luckily they have proper default values.) I then interrupted (instead of booted) into another volume. Because of that, this volume's clock tried unsuccessfully to locate a time server and, by default, resumed ticking from when I left Friday evening! And once it begins ticking, it apparently never checks again for a time server.

When I typed in my RISKS contribution and sent it, it had that Friday timestamp, though it was Monday and I was (correctly) citing a Sunday news article.

--Rodney

Estimating Unreported Incidents

Ken Laws <Laws@SRI-AI.ARPA>
Fri 20 Jun 86 16:21:04-PDT

[In RISKS-3.8, I noted how rarely I get two reports of the same incident, and wondered how many do not get reported at all. PGN]

There is actually a statistical technique (based on the Poisson distribution, I'm sure) for estimating the number of unreported items from the frequencies of multiply reported ones. It was developed for estimating true numbers of Malaysian butterfly species from collected ones, and has recently been used to validate a newly discovered Shakespeare poem from the percentages of words that were used 0, 1, ... times in the accepted Shakespearean literature.
-- Ken Laws

Estimating Unreported Incidents -- and risks of using statistics

Peter G. Neumann <Neumann@SRI-CSL.ARPA>
Tue 24 Jun 86 01:09:31-PDT

Ah, Ken's message brings us to the risks of computer authentication! The poem in question really did not read like authentic "Shakespeare" to me; it seemed vastly too pedestrian, childish, and uncharacteristically repetitive. But then, don't get us started on who actually wrote the works attributed to William Shakespeare. That might be a little risky for this Forum. (However, for some fascinating background, see Charlton Ogburn's book "The Mysterious William Shakespeare -- the Myth & the Reality", pursuing the case that the man known as "William Shakspere" was functionally illiterate, with almost no documents bearing his signature or handwriting and no known contemporary literary activity, and that he could not possibly have written the works attributed to "Shakespeare".) (By the way, I don't think it was Marlowe, Bacon, or -- as Ogburn contends -- Edward de Vere

---

Re: Privacy legislation (RISKS-3.8) and radio eavesdropping

Jerry Mungle <JMUNGLE@USC-ISIF.ARPA>
16 Jun 1986 06:09:22 PDT

Re: Michael Wagner's query about privacy of radio telephone...

[Here are THREE more messages on this subject. Each adds a little more to what Dan Franklin contributed in RISKS-3.10. This time I did not have the patience to edit each one down to its nub, so please read them accordingly... PGN]

For quite a while telephone traffic has been carried by satellite links. It is quite easy to receive such transmissions using nothing more sophisticated than a backyard dish antenna, and the demultiplexing needed to recover a conversation is doable by undergraduate EEs. I believe it is quite illegal to "intercept" phone conversations (or data transmissions via phone lines) in this fashion. However, it is *very* difficult to detect such activities.

I do not believe it should be illegal to monitor ANY radio communication, as the airways are public property. But there seems to me to be precedence for laws regulating reception of radio transmissions (beware, I am not a lawyer).

The risks to computer systems lies in the ease with which data transmitted over phone lines may be intercepted. This relative ease is offset to some degree by the difficulty of finding the particular phone link one wishes to monitor. But, given a reasonable level of support, it should be possible to eavesdrop on conversations/data transmission which one desires to hear. Sales figures, marketing info, experimental data.... lots of valuable data go unencrypted over the phones every day.

Re: Privacy legislation (RISKS-3.8) and radio eavesdropping

Jeff Mogul <mogul@su-shasta.arpa>
17 Jun 1986 1128-PDT (Tuesday)

In RISKS-3.8, ubc-vision!utcs!wagner@seismo.CSS.GOV (Michael Wagner) asks:

> Does anyone have any idea how the last part (radio telephones) could be legally supported in view of other legal freedoms? I thought that one was free to listen to any frequency one wished in the US (Canada too). You don't have to trespass to receive radio signals.

It's been a decade or so since I was familiar with current US communications law (as a licensed Amateur Radio operator, I had to pass several exams covering this sort of thing), but I recall that although there is no prohibition against receiving radio signals, there is a prohibition against divulging what you receive to any other party. Of course, this doesn't apply to all radio services (it's not against the law to reveal baseball scores you heard on an AM broadcast station) and I doubt it's often enforced.

Compare this to what a computer system manager might face when unraveling a mail snafu. I might not be able to avoid seeing the text of an unencrypted message (as I watch packets moving between hosts) but it would certainly be unethical for me to reveal what I saw, or indeed to make any use of it. Ideally, the technology would be such that I could not accidentally see the contents of a message while performing a management function, but in today's world I think the only enforceable prohibition is against divulging or using electronic mail, not against seeing it. (Of course, seeing by means of unauthorized access is also prohibitable.)

-Jeff Mogul

Re: Privacy Legislation (RISKS-3.10)

Jim Aspnes <asp@ATHENA.MIT.EDU>
Mon, 23 Jun 86 11:39:45 EDT

Date: Tue, 17 Jun 1986 00:32 EDT
From: LIN@XX.LCS.MIT.EDU
To: ubc-vision!utcs!wagner@SEISMO.CSS.GOV (Michael Wagner)
Cc: RISKS-LIST:@XX.LCS.MIT.EDU, risks@SRI-CSL.ARPA
Subject: Privacy legislation (RISKS-3.6)

[On the same topic...]

Not true. States routinely ban the use of radar detectors, and that is nothing more than "listening to a frequency."

Most states do not actually ban the use of radar detectors, but rather the operation of a motor vehicle containing one; as I understand it, if you want to sit at home and detect your burglar alarm, you are entirely within the law. There is no constitutional or federal restriction on how states can regulate your driving.

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 13

Thursday, 26 June 1986

Contents

- The Risky Gap Between Two Design Cultures
  Jack Goldberg
- Risks of nuclear power
  Dan Franklin
- Research programs that pay for themselves
  Rich Cowan
- Having an influence from "within the system"
  Rich Cowan
- RISKS in running RISKS -- continued
  PGN and an unhappy Mailer
- Info on RISKS (comp.risks)

The Risky Gap Between Two Design Cultures

Jack Goldberg <JGOLDBERG@SRI-CSL.ARPA>
Wed 25 Jun 86 12:01:12-PDT

Over the centuries of experience in dealing with hazards, mechanical and civil engineers developed a culture of safe design, with principles and practices appropriate to the various kinds of products. This culture was expressed in the design of mechanisms that implemented various safety functions, such as barriers to undesired motion, redundancy in the event of local failures, self-adjustment to losses of tolerance, and so on. For each kind of product, particular mechanisms were developed to accomplish these functions, e.g., pawls, detents, rails, ratchets, fuses.

The advent of computers and inexpensive sensors and motors made possible tremendous economies in manufacture by eliminating all those particular mechanisms and their often costly assembly (consider the dramatic comparison in complexity of mechanism between the original teletype machine and a modern typewriter/printer). Mechanical design of the new systems has been dramatically simplified, and the complex functions, including safety functions, have been relegated to a control program. In a sense, the design is created on a blank slate.

Who creates that design? Generally someone who is a professional programmer, often a novice, who has inherited the culture of that profession. There are many aspects to that culture, but it rarely includes the lore and practices of safe design (and the exposure to the machinery of legal liability) that is the inheritance of mechanical and civil engineers. It is often based on a partitioning of responsibility between the hypothetical (and often anonymous) "customer" and the programmer-supplier, a partitioning that hides the ultimate users from the designer. Also, too often, the programmer's education in matters of the physical world has been compromised by the demands of training for his profession.

Often, the practitioners in the new culture see themselves as generalists, able to solve any new problem, and they move frequently from one application area to another. Consequently, they seldom have the time to study and understand the things that users or designers in a particular field know or assume to be obvious, and so they must imagine and re-invent them. Tragically, those imperfectly mastered things sometime seriously affect safety.

In short, the culture of safety that traditional engineers have expressed in particular mechanisms has been tossed out along with those mechanisms, and is being re-discovered, painfully, by a new generation of designers that has no connection with the traditional culture. In this light, risks arising in contemporary computer-based system design may be seen as a consequence of a gap between two design cultures. The gap is both generational and professional; there are many safety engineers in industry, but they and programmers speak different languages.

In a different context, awareness of the loss of knowledge by experts in various practices, due to their lack of replacement in the work force, has stimulated some computerists to try to capture that knowledge. How well they are doing that is another matter, but it may be that some conscious gap-bridging between the cultures would save the world some amount of misfortune and misery.

Jack Goldberg, SRI International

Risks of nuclear power

Dan Franklin <dan@bbn-prophet.arpa>
Tue, 24 Jun 86 14:03:42 EDT

TMPLee@DOCKMASTER.ARPA discusses nuclear energy vs. solar energy and "taking all the risks into account". The risk he is primarily concerned with is the risk of falling off a solar energy device while cleaning off the snow.

If we are going to take all the risks into account, let's face it: the risks to those involved in the actual energy production are simply insignificant in the debate on nuclear vs. other forms of energy. That debate focuses almost entirely on the risks to innocent bystanders. These are the risks that always matter most, precisely because people do not willfully undertake them, but rather end up subjected to them, and people are not willing to be

subjected to nearly as much risk as they are willing to *decide* to take,
or let others decide to take.

The fundamental political problem of nuclear power is that it has a small probability of being disastrously more injurious to bystanders than any other form of power generation except dams. (Solar power satellites which deliver their power by microwave are a future contender.)

TMPLee's mention of low-level radiation emitted by coal-fired plants is, of course, directly relevant to this issue. But in the wake of Chernobyl, as in the wake of TMI (and in the wake of Pilgrim's safety problems...), the small probability of disaster clearly needs to be discussed.

Dan Franklin

Research programs that pay for themselves

Richard A. Cowan <COWAN@XX.LCS.MIT.EDU>
Thu 26 Jun 86 00:08:21-EDT

Let me add a few comments to Bob Estell's point #6:
> "Going to the moon in the '60's cost the USA nothing!...
> The DIFFERENCE between tax dollars paid by those wearing pacemakers, and
> the "aid to their families" that would have been paid had those heart
> patients died or been disabled, is more than $25 billion."

There are two problems with these types of conclusions. First of all, there are plenty of big non-space or non-military government programs that we could spend our money on that are equally likely to have spinoffs; there must be a reason why SDI should be built rather than these projects. But the classification barriers of SDI will inevitably reduce spinoffs. Not only that, but some things in SDI will certainly be useless commercially. Pacemakers don't need to survive nuclear explosions.

Secondly, any government program has an opportunity cost which is not factored into your calculation: when we devote scientific resources to the private sector, we lose out on the benefits we would have gained if those resources weren't used up by the government. An example is mentioned in a May issue of the weekly trade paper "Electronics News": a Japanese witness at some hearings on US competitiveness points out that the United States spends hundreds of millions on high-strength, lightweight carbon materials for aircraft wings, while the Japanese developed the same materials very cheaply for golf clubs and tennis racquets.

Are there things which we could use more than we could use SDI? Are there other government expenditures (perhaps national health insurance) that would REALLY cost nothing? Well, I recently heard that aside from public police forces and the military, about $300 billion per year is spent on security (including locks, alarms, etc.) To get an idea where this comes from, consider that MIT's police force costs a couple million, and all universities put together must spend about $1 billion.

Now if businesses instead spent $100 billion of this money on raising the minimum wage $2, spent $50 billion on reducing unemployment by reducing the work week, and $20 billion went to the government to improve housing programs and public facilities to keep young people occupied, then perhaps the need for so much security would be reduced, because the root causes of crime would be diminished. It would therefore "cost nothing" for the private sector to divert $170 billion of its security bill and improve the social stability and welfare of the country. The problem with such a plan is that the benefits come only in the long term; only the greater short term costs are seen on corporate balance sheets.

-rich

Having an influence from "within the system"

Richard A. Cowan <COWAN@XX.LCS.MIT.EDU>
Thu 26 Jun 86 00:11:07-EDT

And now a few comments on Bob Estell's point 10 on working for SDI:
> "But if we do take the opportunity, then we can use the
> managers' short term interests to an advantage; i.e., we can honestly say
> that "Star Wars" [R2D2 et al] is not possible today; and then diligently
> work to produce what is reasonable."

You have here touched upon what I believe is -- more often than not -- a delusion: that it is more effective to work within the system to change it than to protest it from without. In this case, working within the system means working on Star Wars to demonstrate part of it to be feasible or infeasible.

There are several problems with this. First, within a large institution you may be isolated from resources, or a diversity of viewpoints needed to make an impartial decision. This is less true with Star Wars than with other programs because there's lots of mainstream publicity. It is also less true in a university than in a defense contractor.

Second, and more importantly, what an engineer says is likely to get manipulated for political reasons -- like the ignored warnings before the space shuttle disaster. If of 10,000 engineers working on SDI, 5000 include negative critical material in their research reports, and the other 5000 are completely uncritical of SDI, what do you think Congress will hear? Well, I can guarantee that they will hear mostly glowing reports about research progress from upper-level managers and lobbying organizations of the companies doing SDI research. If your strategy to change things is to become one of those upper-level managers, you may have to compromise your values to achieve promotion, and temper your criticisms to avoid losing "credibility" once you get there.

Yet Congress is hearing the other side on SDI. How? Because engineers are not relying on the companies they work for to communicate their insight. They are going outside the normal channels of communication -- like the 1600 scientists working at government labs who recently petitioned Congress to curtail SDI spending. And ultimately, communicating one's concerns directly to people in the community is necessary.

What is unfortunate, and I believe dangerous in a democracy, is that people working for the government are afraid of speaking out on public policy issues for fear of reprisal. The recent statements by Undersecretary of Defense for Research and Engineering Donald Hicks may have heightened this fear. Fortunately, the Pentagon has recently dissociated itself from Hicks' statements. (Science, May 23)

-rich

Returned mail: Service unavailable

Mail Delivery Subsystem <MAILER-DAEMON@nprdc.arpa>

[One of the greatest annoyances in running a large mailing enterprise such as RISKS is fielding the incessant net-barfs, including having my mailbox cluttered with multiple copies of the Forum on net addresses that don't work now and then. (Some mailers that keep retrying periodically, and send back advisories each time.) Here is a fine example -- which of course more generally represents another type of risk in distributed systems. PGN]

[FOOTNOTE: The more general problem of copious rejected mail would be an order of magnitude worse if we went to individual messages rather than the current digest format. (I now have requests from BITNET and USENET to do send out undigestified messages, and would love to let them do the undigestifying. Perhaps more regional reforwarding centers would minimize rejects and reduce mailing list maintenance substantially. But, I nevertheless get mystery rejection notices for people not even on my list, because of redistribution problems elsewhere.)]

```
  ----- Transcript of session follows -----
<>> DATA
<<< 554 <malloy>... Mail loop detected
<>> QUIT
<<< 554 sendall: too many hops (30 max)        [... but quite a brew-haha!]
554 <malloy@hull>... Service unavailable: Bad file number

  ----- Unsent message follows -----
Received: from pacific.ARPA by nprdc.arpa (4.12/ 1.1)
   id AA00971; Tue, 24 Jun 86 03:04:28 pdt
Received: from hull.aegean.arpa (hull.ARPA) by pacific.ARPA (4.12/4.7)
   id AA11313; Tue, 24 Jun 86 03:04:01 pdt
Received: from nprdc.arpa (aegean) by hull.aegean.arpa (2.2/SMI-2.0)
   id AA14974; Tue, 24 Jun 86 02:50:15 pdt
Received: from pacific.ARPA by nprdc.arpa (4.12/ 1.1)
   id AA00967; Tue, 24 Jun 86 03:04:07 pdt
Received: from hull.aegean.arpa (hull.ARPA) by pacific.ARPA (4.12/4.7)
   id AA11309; Tue, 24 Jun 86 03:03:40 pdt
                  [... many hops omitted ...  I think you get the idea.
```

Notice the clock drift while you're at it.]

Received: from hull.aegean.arpa (hull.ARPA) by pacific.ARPA (4.12/4.7)
  id AA11281; Tue, 24 Jun 86 03:01:05 pdt
Return-Path: <NEUMANN@SRI-CSL.ARPA>
Received: from nprdc.arpa (aegean) by hull.aegean.arpa (2.2/SMI-2.0)
  id AA14942; Tue, 24 Jun 86 02:47:19 pdt
Received: from pacific.ARPA by nprdc.arpa (4.12/ 1.1)
  id AA00935; Tue, 24 Jun 86 03:01:10 pdt
Received: from nprdc.arpa (aegean.ARPA) by pacific.ARPA (4.12/4.7)
  id AA11271; Tue, 24 Jun 86 03:00:39 pdt
Received: from SRI-CSL.ARPA (sri-csl.arpa.ARPA) by nprdc.arpa (4.12/ 1.1)
  id AA00926; Tue, 24 Jun 86 03:00:30 pdt
Date: Tue 24 Jun 86 01:41:53-PDT
From: RISKS FORUM   (Peter G. Neumann, Coordinator) <RISKS@SRI-CSL.ARPA>
Subject: RISKS-3.12 [...]

[But, gee Mr. Wizard, it worked just fine on the previous issues!]

Search RISKS using swish-e

Report problems with the web pages to the maintainer

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 14

Friday, 27 June 1986

Contents

- A Personal View on SDI
  Harlan Mills
- Privacy legislation (RISKS-3.10)
  Jerome H. Saltzer
- Risks in burning wood
  Mike McLaughlin
- Mailer explosion
  Sean Malloy
- Info on RISKS (comp.risks)

A Personal View on SDI from Harlan Mills

Peter G. Neumann <Neumann@SRI-CSL.ARPA>
Fri 27 Jun 86 13:35:07-PDT

[The following note has been circulated privately by Harlan Mills, noted practitioner of structured programming and other software engineering techniques, and is included here with his permission. PGN]

Two of my friends, whose intelligence and integrity I respect and admire greatly, namely David Parnas and James Horning, have stated their belief that the SDI concept is impractical. At the same time other groups of scientists and engineers, from dozens to hundreds to thousands are declaring their opposition to SDI on various grounds from infeasibility to conscience. Yet, we do not seem to find comparable groups of scientists and engineers on the pro side of SDI in public forums. Is it because there is no pro side? Or is there some other reason? I think there is another reason.

First, there are many scientists and engineers actively working on SDI research. Does that mean they are for SDI or are simply hypocrites? I think for most of them that neither is the case. There is another reason possible. I believe it is the case with me.

I personally do not know enough to be for or against SDI. But I do know enough to want our country to be strong in technology. As a citizen, I depend on our system of government, and particularly our Congress, to decide about SDI.

I regard SDI as a political question that will be ultimately settled in our political system by the 525 members of our Congress. I trust them to make the wisest disposition possible of this question. It seems too complex a qustion to settle on a simple up or down vote. It will take time, experience, and reflection to progressively deal with it. Much of that experience and reflection will be political and diplomatic; some of it will be military and technical in nature. I believe the intent of most scientists and engineers working on SDI is to explore the technical side intelligently enough to provide the widest range of options possible for the political and diplomatic side.

In order to pursue the SDI question, the administration, particularly the military, must organize a substantial and serious effort that itself involves a narrower form of political effort. It must advocate a position and lobby Congress for the opportunity to pursue SDI military and technical research in a responsible way. But I do, indeed, believe that members of Congress, with the facts, the checks and balances of our political system, and constitutional guarantees (e.g., a free press) will resolve the question of SDI intelligently in due course and process.

So I regard the positions of my friends Parnas and Horning, and of many other scientists and engineers, as thoughtful and courageous acts of technical or political conviction. In particular, Parnas and Horning are expert witnesses in computer science and software engineering. People in the administration and members of Congress should and do listen to them. In matters of theory in computer science or software engineering, I have never had an occasion to differ or disagree with either of them. But I do not always agree with their extrapolations into engineering expectations in large systems such as required by SDI.

In the first place, I believe it is somewhat misleading to convert the problem of SDI feasibility into the question of software perfection. The problem is deeper than software. The recent shuttle tragedy reminds us that any man-made system can fail for many reasons beside software. So the problem is even worse than simply software. The best man can do in any physical system is to reduce the probability of failure to low levels, not to zero. If the hardware fails more often then the software, it is wiser to improve the hardware even though the software is not perfect.

In the second place, I believe that engineering expectations and achievements in large systems depend as much on the checks and balances of good management processes as on engineering theory. We never get away from the fallibility of people, but we can reduce the fallibility of organizations of people below the fallibilities of their individuals. And with sound engineering theory, there is no real limit to that reduction in fallibility of organizations. For me, they key is the combination of sound engineering theory and good management process -- both are necessary and neither is sufficient.

So my extrapolations into what is possible for SDI software are more open ended than those of Parnas or Horning. But, as Parnas and Horning both suggest, we surely will not get there doing business as usual in the DoD software acquisition process. Thus, as with the Congress, I expect DoD to rise to the occasion as the needs arise. After all, it's our DoD, as well as our Congress.

In another era, in the late 40's I was involved in a losing cause on the issue of "One World or None." As a student, I was convinced by the arguments of my elders that atomic theory should be declassified and that the U.S. should lead the way with an open science policy throughout the world. The science world was split then -- Niehls Bohr on one side, Edward Teller on the other (and Robert Oppenheimer, I think, caught in the middle). But, of course, the cold war and Korea settled things irreversibly. In spite of the excesses of a few individuals, I believe our Congress and administration came through that period as well as possible in steering a science policy course. I was personally disappointed in a dream of open science and abundant peace, but I do not see how it could have been pulled off if our government could not see how.

That is how I look at SDI. I would like to help my country be strong in science and engineering. The adminstration and the military are agents of the country in that endeavor. But, I depend on the Congress to make the final, collective, decisions, in how to best reflect that strength for peace in political, diplomatic, and military matters.

However, as events unfold and we all learn more, both about SDI needs and engineering theory, if I come to the same belief as Parnas and Horning, you can be sure that I will join them, and try to bring my opinions to the administration and Congress, too. I want to be on the right side, whether it loses or not!

Harlan Mills

Privacy legislation (RISKS-3.10)

Jerome H. Saltzer <Saltzer@ATHENA.MIT.EDU>
Fri, 27 Jun 86 15:16:13 EDT

The reported privacy legislation proposal for radio-based telephone conversations is quite analogous to some of the proposals that circulated for several years around the cable and satellite TV industry. In that case as well as this, technology bluffing is dominating the conversation. The overall scenario is that economic interests are claiming that technology can't supply privacy economically, so draconian laws are the best way to proceed. Responsible engineers should object to this line of reasoning whenever they notice it being misused.

Since in-the-clear radio communications are trivially, even accidentally interceptable, the public interest requires that the first avenue to explore in protecting them be narrowly technological (scrambling) rather than broadly targeted legal approaches that can have surprising side effects on the bill of rights. But commercial interests that don't want to think about extra costs or delay in getting to market use technological intimidation to produce public positions that scrambling is too expensive.

The cable and satellite broadcast communities have come to realize that laws don't help as much as they hoped and they have to scramble anyway. It would be nice if we could somehow get that fact across to the legislators who are being bamboozled by the cellular telephone business.

The worst part about passing a law to cover for temporarily missing technology is that when the technology to solve the problem does arrive, the laws don't magically disappear; they stick around, forgotten, to cause trouble and surprises later when an enterprising District Attorney discovers they have undreamed-of possibilities.

A related comment on banning listening said. . .

> Not true. States routinely ban the use of radar detectors, and that
> is nothing more than "listening to a frequency."

States often legislate things that wouldn't pass constitutional muster; this is an example that at least some legal specialists identify as unlikely to stand up. The word around here is the real challenge to radar detector bans is awaiting the first time that the state of Connecticut tickets F. Lee Bailey.

Jerry Saltzer

Risks in burning wood

Mike McLaughlin <mikemcl@nrl-csr>
Fri, 27 Jun 86 11:21:19 edt

Risks has carried a lot lately regarding the risks associated with nuclear energy. Some discussion has compared nuclear with coal and hydro. The emphasis has been on "disasters," such as Chernobyl or dams breaking.

May I respectfully submit that not all disasters are sudden.

Wood smoke is a pollutant. It may smell nice (except for poplar and a few others), but if you burn enough of it, nasty things happen.

Coal smoke is a pollutant. It never smells nice, and it makes for acid rain and other nasty things. These nasty things are slow, but some of us recognize the long term effects of generating power through coal as an ecological disaster.

Most natural hydrocarbon combustion byproducts (excuse me, "smoke") also contain carcinogens. They are as effective at producing cancer as alpha, beta, gamma, and all those other funny names. Just different cancers.
I see no value in having any cancer, different or not.

In an attempt to tie this to computers somehow, so that PGN will not toss this in his bit bucket:

Will some reader please gather a creel of Crays and compare the long-term hazards to the populace, Sialis sialis and Cornus florida of nuclear pollutants (sudden or slow) vs. hydrocarbon pollutants (sudden or slow) while holding Terra's total energy demand as a constant?

Thank you.

Mailer explosion

Sean Malloy <malloy@nprdc.arpa>
Thu, 26 Jun 86 06:50:03 pdt

I'm sorry about the explosion of the mailer demons here. At NPRDC, we have a network consisting of two VAXen, eight or nine Sun workstations, and a couple of PCs and ATs, all EtherNeted together. The mail program was recently brought up on the Suns, and it was suggested that people wishing to receive their mail on the Suns rather than on PACIFIC (the VAX our code has primary accounts on) should put .forward files in their home directories on PACIFIC, which would cause mail sent to <username>@pacific to be forwarded to a system specified in the .forward file.

So I made a .forward file, and expected my mail to be forwarded from malloy@pacific to malloy@hull. But I hadn't expected that a network mail alias simplification would blow my mail all over creation. To simplify maintaining the mail alias file on the Suns, the file /usr/lib/aliases on PACIFIC gets copied to the Suns whenever it is changed. This means that the Suns think my mail address is malloy@pacific.

As a result, any mail coming in between Friday (6/20) morning when I set up the .forward file, to Monday morning when I deleted it because it wasn't working right (one of my coworkers mentioned losing mail to me) was received by pacific, where the mailer-demon read the .forward file, and sent it on to malloy@hull. Hull received the mail, checked the /usr/lib/aliases file, and sent it back to malloy@pacific. Twenty-nine loops later, the mailer-demon explodes, and my mail gets thrown back at whoever sent it.

Sean Malloy (malloy@pacific)

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 15

Sunday, 29 June 1986

Contents

- A Personal View on SDI from Harlan Mills
  Herb Lin
- Having an influence from "within the system"
  Herb Lin
- Re: Research programs that pay for themselves
  Rich Cowan
- Text Scanners
  Fred Hapgood
- Info on RISKS (comp.risks)

A Personal View on SDI from Harlan Mills

<LIN@XX.LCS.MIT.EDU>

Sun, 29 Jun 1986 06:12 EDT

On the whole, I am touched by Harlan Mills' remarks. But I am bothered by two things. He says that

I [Mills] regard SDI as a political question that will be ultimately
settled in our political system by the 525 members of our Congress.
I trust them to make the *WISEST* disposition possible of this question.

I depend on the Congress
to make the final, collective, decisions, in how to *BEST* reflect that
strength for peace in political, diplomatic, and military matters.
[Emphasis added by me]

These comments reflect a trust in a rational process of government that I wish I could share; it almost sounds as though he believes that whatever decision the Congress makes will be right *by definition*. I have seen too many instances in which Congress manifestly did NOT do the right thing to believe in their collective wisdom. The nature of a democratic system forces me to *abide* by their decisions, but that is not the same thing as approving of them or believing in their wisdom. (On the other hand, I would not trade democracy for anything else.)

At a somewhat more fundamental level, he states that

> .. it is somewhat misleading to convert
> the problem of SDI feasibility into the question of software perfection.
> ... The best man can do in
> any physical system is to reduce the probability of failure to low levels,
> not to zero.

The latter statement is a position with which all TECHNICAL analysts agree: a perfect system is impossible. But the POLITICAL debate has been cast in terms of "Do you want to defend yourself or not?", "eliminating (NOT reducing) the threat of nuclear ballistic missiles" and "the immorality of threats to kill innocent civilians".

The technical analysis of the political questions posed above is absolutely clear, and is that it is impossible to develop technology that will allow us to get rid of offensive nuclear weapons and shrug off nuclear missiles should they happen to be launched our way). Technical analysts then debate the technically more interesting question of what CAN be done, in which case Mills' comment that

> ... the intent
> of most scientists and engineers working on SDI is to explore the technical
> side intelligently enough to provide the widest range of options possible
> for the political and diplomatic side.

makes a great deal of sense.

But SDI supporters in the political arena find THIS question much less interesting. The support that SDI garners from the population at large, and indeed from those that push it arises from the fact that defense against ballistic missiles is a truly revolutionary possibility, that will result in a military posture that is qualitatively different from that which exists at present. It won't, as SDI supporters admit when pushed; they say defenses will enhance deterrence, and that we will still have to accept societal vulnerability and to rely on the threat of retaliation to deter Soviet attack.

Looking at the question from another side, all technical analysts agree that it is possible to build SOMETHING that sometimes does some fraction of what you want it to do, and the interesting technical questions are what is the nature of this something, what will it be able to do, and how often can it do it. But the political debate is cast against the backdrop of technology that is capable of meeting a certain absolute level of performance, and a rather high one at that. The technology to do THAT is much more demanding -- if the level of performance is societal perfection, then it's not reachable at all. The political proponents try to have it both ways; they want the political support that comes from belief in the feasibility of this very demanding technology, and they try to deflect technical criticism

of this political position by saying the question is one of
discovering what technology can do.

Thus, until the broader political debate can be recast in terms of the desirability of IMPERFECT defenses, and SDI supporters concede POLITICALLY that defenses will not do what is being claimed for it, technical analysts, in my view, are fully justified in pointing out that perfection is not possible. When SDI supporters make this concession, the perfect defense issue will become a dead horse politically as well as technically, and we can all go on to talk about more interesting things.

Having an influence from "within the system"

<LIN@XX.LCS.MIT.EDU>
Sat, 28 Jun 1986 17:52 EDT

> From: Richard A. Cowan <COWAN>
>
> You have here touched upon what I believe is -- more often than not -- a delusion: that it is more effective to work within the system to change it than to protest it from without.

Without addressing the specific merits of doing SDI work at this time, I think this statement needs qualification.

There is a role for people outside the system. There is also one for people inside the system. Activists are necessary to bring political pressure. But they have to have some technical credibility. As bad as things are in government now (with people believing in the Tooth Fairy,.. excuse me, I meant perfect ballistic missile defense), there is only minimal support for other things that other people would also like to have -- teaching creationism in the schools for one. The reason is that there is NO serious scientific opinion that creationism has any literal validity at all. I can assure you that if there were, the battle to keep creationism out of the textbooks would be a lot more difficult to fight.

Technical credibility is not the same thing as being "inside the system". But "the system" does many things, some of which are probably right, and others wrong. But should that mean that people should give up on the whole thing? Some of the most effective critics of the system are those who have extensive experience in it -- Richard Garwin comes to mind as a prime example. His effectiveness comes about because he knows what he is talking about, and it is hard to imagine that he could have developed his expertise had he remained forever outside the system. By contrast, Kosta Tsipis -- while he has made a rather significant name for himself in the public domain -- has been identified in most of the public debate that I have heard as a flake who instinctively knee-jerks against US defense; Tsipis has never been part of "the system". (This is not to make a judgement about the quality of Tsipis' work.)

Then why doesn't the system stop doing silly things? I guess the answer has to take the form -- if you think things are bad now, just imagine how much worse they would be without the likes of Garwin. While being technically right doesn't necessarily mean that your position will win, being technically wrong is often the kiss of death.

Re: Research programs that pay for themselves

Matthew P. Wiener <weemba@brahms.berkeley.edu>
Sun, 29 Jun 86 02:47:49 pdt

I'd like to add a small comment to Richard Cowan's remarks.

One concern about SDI spinoffs is that DoD gets to choose some of them. I wonder if, for example, we are going to see more incidents like the ASATing of Solar Max--a fully working scientific satellite whose routine operating grant renewal was turned down last summer to provide a suitable test target.

ucbvax!brahms!weemba   Matthew P Wiener/UCB Math Dept/Berkeley CA 94720

Text Scanners

"Fred Hapgood" <SIDNEY.G.HAPGOOD%OZ.AI.MIT.EDU@XX.LCS.MIT.EDU>
Sat 28 Jun 86 06:33:34-EDT

The archetypal computer risk is of course unemployment. With regard to this issue, does anyone know what sort of inroads page and form scanners are or are not making into the data entry industry, and what features are pacing or retarding penetration into that market? Or would anyone have any suggestions of whom I might call to find out more?
[Please respond privately to Fred unless your response has RISKS-related implications. PGN]

Search RISKS using swish-e

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 16

Monday, 30 June 1986

Contents

- Chernobyl (a suprise to the Soviets)
  Martin Minow
- Airwaves & Security (2 Subjects)
  Richard S. D'Ippolito via dhm
- Interesting Technical Questions (originally SDI)
  Martin Moore
- Info on RISKS (comp.risks)

Chernobyl (a suprise to the Soviets)

Martin Minow, DECtalk Engineering ML3-1/U47 223-9922 <minow%pauper.DEC@decwrl.DEC.COM>
30-Jun-1986 1510

From the Danish newspaper Information, May 31, 1986.

Soviet Union
   Ove Nathan: Chernobyl Totally Choked the Leaders

The Danish atomic physicist and rector for Copenhagen University, Ove Nathan, who is currently attending a conference on atomic weapons in Moscow, said Friday [May 30] in an interview with Swedish Broadcasting that an intensive discussion is going on behind the scenes in the Soviet Academy of Sciences.

According to Ove Nathan, the accident at Chernobyl totally choked the politicians in charge of the Soviet Union. They had never imagined that something similar could have occurred.

Ove Nathan has spoken with several members of the Soviet Academy of Sciences who said that the mathematical calculations they used in their probability computations were completely incorrect. These must be revised, and possibly also the decision to locate nuclear reactors in or near densely populated areas.

"The new thing is that they openly admit that they do not know how they will handle the situation after the accident. They say that is extremely complicated, nothing can be taken for granted, and there are no sure factors one can rely on. Every day brings a new surprise."

Professor Nathan suggests that this is a situation that is completely un-Sovietic. This is the first time in the Soviet history that the elite in the Soviet Academy of Sciences admit that they don't have firm ground under their feet.

Ove Nathan believes, that the most serious consequence of the Chernobyl catastrophe will be an increased demand in the Soviet society for open information from the government.

Translated by Martin Minow

[The Danish original of the text that I translated as "the mathematical calculations they used in their probability computations were completely incorrect" is "den matematiske kalkyle, man har anvendt i sine sandsynlighedsberegninger, var helt fejlagtige" -- I don't have a dictionary so I'm not quite certain my translation was completely correct.]

Airwaves & Security (2 Subjects)

<dhm@sei.cmu.edu>
30 Jun 1986 15:20-EDT

[This message is being forwarded for Richard S. D'Ippolito (rsd@sei.cmu.edu) whose machine does not yet have ARPAnet access; replies temporarily to dhm@sei.cmu.edu]

AIRWAVES

It seems to me that what's been missing in the debate on Airwaves/Privacy is that 'public' ownership is being erroneously equated with 'free access'. We certainly pay camping fees at public parks and tolls on some public roads. Public ownership of the airwaves (essentially nothing real) means simply equal access under the same set of government (public) rules and regulations so that no group is denied access for discriminatory (in the constitutional sense) reasons. Now then, why should a business expect to have its product stolen, which is essentially what is happening? And why can't they protect their normal interests, i.e., proprietary information, with whatever security deemed necessary and have the government back them up (with laws and penalties) just as they do with communications through the mails -- another 'publically owned' and equally accessible enterprise? And by the way, your rights in this state (PA) in public parks are considerably restricted from what they are on your own property -- no firearms, alcohol, pets, or explosives. I can't feel sorry for those who want to steal a service.

SECURITY

Mr. Richard Cowan has presented what I think to be a commonly held but misconceived argument on security, locks, and crime. It is not the proper duty or function of business to reduce the causes of crime by paying unrealistic wages or creating unnecessary jobs. Some people are thieves, period, not because they are poor or unemployed. And, as long as there is one left, all prudent people will want locks. Please, let's skip the sociological arguments in the discussions of SDI. [Disclaimer: For those who do not know (most of Pittsburgh doesn't yet) the SEI is not involved with SDI, nor do we write war (or any) software here -- no flames, please.]

The SDI should be evaluated on several, I believe, criteria. Please let me try to be brief and state several assumptions (which not all of us may hold):

() We have a defense need (implicit function of the government).
() The perfect defense is one that is never tried.
() The Soviet Union is our strongest enemy.

Given these, we can view the SDI in several ways (sorry to condense):

() If the Soviets are against it, it must be good for us, i.e., it's a political diversion and keeps them from spending more time on sorry ventures like Afghanistan.
() It doesn't have to work -- it's successful if no enemy tests it.
() If it causes our enemies to spend a lot of time and resources to match it, then the diversion of their resources from their people can de-stabilize the government through the rise of dissent and unrest.

Now, don't we need to include issues like that in the evaluation of any defense? I'm certainly as unhappy as anybody about wasted tax dollars, as I pay to many of them now. Also, I would like to live in a peaceful world (read risk-free), too, but it just isn't going to happen. I would like all engineers (I'm one) and scientists to take the high side of the debate to the public -- that we work our butts off to make things as risk-free as possible and that we are willing to discuss and quantify (where possible) the magnitude and probabilities of the risks.

In Great Britain, they talk about these things to the public all the time. Here, only the insurance companies know. For example, in building a chemical plant, the calculations of the magnitudes and probabilities of a life-injuring or -destroying accident and the resulting cost (yes, they put cold numbers on them -- your medical insurance company already has the value of your arm listed) is factored in along with all the other costs to determine the proper design and location of the plant in economic terms.

It is totally unrealistic for us to put infinite values on human lives (I didn't say life) because that's when we conclude that everything must be perfect and risk free. A perfect example of this kind of reasoning can be seen in the FDA's treatment of hazardous substances. Have you notice that the allowable limits of these substances always decreases to the limits of measurability as new measuring instruments are devised, even in the absence of direct risk at those levels which are now orders of magnitude below the levels accepted as harmful? Where do we stop? In more concrete terms, I was unable to attend a lecture on this subject: Is a program with a known and predictable error rate of one wrong answer in 10,000 executions useless?, but the subject did intrigue me.

--- Richard S. D'Ippolito (rsd@sei.cmu.edu)
Software Engineering Institute
Carnegie-Mellon University

Interesting Technical Questions (originally SDI)

<mooremj@eglin-vax>
0 0 00:00:00 CDT

> Looking at the question from another side, all technical analysts
> agree that it is possible to build SOMETHING that sometimes does some
> fraction of what you want it to do, and the interesting technical
> questions are what is the nature of this something, what will it be
> able to do, and how often can it do it.

...and how much will it COST? Not only in money, but in people, raw materials, other resources, etc. This is a fundamental question in ANY engineering effort.

Martin Moore (mooremj@eglin-vax.arpa)

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 17

Thursday, 3 July 1986

Contents

- How Much Computer Literacy Is Enough?
  JAN Lee
- Working within the system
  Rich Cowan
- Re: [Airwaves &] Security -- SDI
  Herb Lin
- Complex issues, complex answers
  Bob Estell
- Politics and Engineering Practice
  Seifert
- Multiple copies of RISKS-3.16
  Kenneth Sloan
- GTE Sprint billing problems
  Chuck Weinstock/Lee Breisacher
- Info on RISKS (comp.risks)

How Much Computer Literacy Is Enough?

<JANLEE%VTCS1.BITNET@WISCVM.ARPA>
Wed, 2-JUL-1986 11:46 EDT

I would like to open a new area for discussion, that I hope can involve three elements of the audience: educators, workers, and philosophers, but hitting at what I believe to be a fundamental element of the "Risks to the Public" concept. It is the area of teaching programming.

Over the past several years there has been a salutory movement in the presentation of first course material away from a course in the syntax of BASIC (etc.) to a course which is now entitled "Computer Literacy". There are numerous textbooks available (25 as of my count published since last Fall alone!) and the topics seems to fall into four basic areas: (1) An overview of what a computer is -- including hardware and software, (2) An excursion into the applications of computers in various fields (which can be tailored to specific student's interests), (3) The social impacts of computers on the world (hopefully including something about risks), and (4) Exposure to some elementary activities such as Word Processing, Spreadsheets, Graphics and/or Data Bases. This organization I support strongly for those for whom this is likely to be the only course they will ever take in this area, and it's not bad also for those who might go on and take a programming course later -- at least they get the background needed for a better understanding of the issues.

NOW FOR MY PROBLEM: We have taught such a course for about four years (since the advent of the PC) and have been pleased with the results, one of which is to strip these students who merely need an exposure to the field out of the later programming courses. HOWEVER, in the normal review for a new course, we were refused approval to continue offering this course unless we included "real" programming. Many departments on campus want to have their students only take one CS course and to be able to program (mostly in BASIC) problems in application areas afterwards.

To do a plausible job of teaching programming (to my way of thinking) requires preparation in the methods of problem solving first and a good grounding in the development process afterwards. Without cutting out the guts of a literacy course, I estimate we have 3-4 weeks (9-12 class periods) to do all this. These students are going to go out and write programs which put people at risk -- dieticians, agriculturalists, etc.

I am refusing to offer this course since I do not believe that I can cast out into the field a group of students whose grasp of the problems of programming are insufficient to protect themselves (and others) against errors. So someone else will teach it!

NOW FOR MY QUESTION: How little can we get away with in preparing students to use the computer for problem solving and not put their eventual clients at risk?

JAN

Re: Working within the system

Richard A. Cowan <COWAN@XX.LCS.MIT.EDU>
Thu 3 Jul 86 21:07:12-EDT

As Herb Lin pointed out, my statement about working within vs. working outside the system had problems. First of all, I unfortunately implied (but did not mean) that "people should give up on the whole thing <lin@xx>"; in fact, I believe that it is almost always possible to work within the system to change it! I think most people can have a significant, visible effect!

The problem is that many people define "working within the system" in a narrow, technical or traditional sense which may blunt or negate the impact they COULD have. Since the nature of our work and the prevailing modes of communication are set up in a compartmentalized fashion to reinforce "the system," one must sometimes circumvent those normal channels to produce change. People are deluded only if they think change will occur through "business as usual."

Although "working outside the system" (and I did not mean violence, as Mr. Jong of Honeywell assumed) sometimes is necessary, organizing a peaceful, but active protest towards a goal may divide people over the goal, alienate those who disagree, produce an institutionally funded backlash, and discourage supporters if it is unsuccessful. Instead of demonstrating, individuals can try to change the CLIMATE in which group positions are formed FROM WITHIN THE SYSTEM, just by banding together in small groups to develop arguments that challenge the standard corporate line.

STRATEGY:
One possible strategy for changing the climate from within is to try to MAKE IT ACCEPTABLE for the head of your company/institution to publicly air your concerns. Although some business leaders may already have strong contrary views, and be impossible to convince, a surprising number may already agree with you -- but remain silent for they lack a support group to give them evidence and confidence.

EXAMPLE:
The president of MIT recently criticized federal research priorities -- 75% military funding of R&D -- in a public speech (Science June 13, 1986, p. 1333). Two things had to happen for him to do this: a) students gave him information documenting these trends and b) people within the upper eschelons of MIT began talking about the issue after it was raised by faculty and students.

This may not seem very significant, but such criticisms are rarely voiced by the heads of US institutions highly dependent on military funding. This sends a signal to all kinds of observers, including policymakers, that the "establishment" is changing course. It also sends a signal to management/professors and workers/students (when the position is reported in the company paper, for example) that makes it easier for them to discuss the same issues.

If 100 additional university and corporate executives were to each be persuaded by the actions of a few people in each institution to make statements on topics generally excluded from public debate, I believe a significant portion of the "consensus" for US domestic and foreign policy would erode. (i.e. imagine what would happen if several corporate executives felt free to voice opinions such as "a foreign policy which makes friends of thousands and enemies of millions does not seem to make good long-term sense" or "certain fields get more research funding than can be efficiently spent.")

WHERE YOU CAN DO IT:
Certainly professional societies and conferences provide a perfect medium for high tech people to raise such issues, thereby making it "acceptable" for others in the profession to have the same concerns. Even a lowly 23-year-old student like myself can have an enormous impact merely by clipping articles for professors or administrators whom I know are concerned but lack the time to get in touch with

activist groups or track down references. Given a few good references, these people won't hesitate to incorporate such ideas into their conversations or speeches, or to express them to people higher in the chain of command. When leaders are concerned, the mainstream press will be more inclined to investigate the issue. When they do, the non-activist public follows.

Since economics necessitates that most people must remain within the system, those people may as well try to make people within existing institutions more open to change. The political role of institutions (especially the leaders) in setting the tone for debate must be held accountable to someone -- why not the employees? Think globally, act locally. People must insist that the meaning of "service to one's institution" be redefined so that duties besides "maximizing its profit in the short term" are included. Otherwise solutions embodying these concerns (i.e. economic conversion) will always appear radical and be immediately dismissed before they reach the public eye.

-rich

Re: [Airwaves &] Security -- SDI

<LIN@XX.LCS.MIT.EDU>
Thu, 3 Jul 1986 10:39 EDT

> From: dhm at sei.cmu.edu
> The SDI should be evaluated on several, I believe, criteria. Please let me try to be brief and state several assumptions.
>
> () We have a defense need (implicit function of the government).
> () The perfect defense is one that is never tried.
> () The Soviet Union is our strongest enemy.

These assumptions follow from another, and in my mind, more basic premise: we want to maintain our way of life free from external coercion. This more basic premise can lead to your set of assumptions, or to different sets of assumptions. For example, it could lead to the assumption that a reduction in tensions is a sensible thing to do, which is not mentioned in your set. Of course, I don't think you intended your list to be complete, so I am just adding to it.

> Given these, we can view the SDI in several ways: (condensed)
>
> () If the Soviets are against it, it must be good for us, i.e., it's a political diversion and keeps them from spending more time on sorry ventures like Afghanistan.

Maybe true and maybe false. If you are my enemy, and you start drilling a hole in your side of the boat, I'm sure going to start complaining. I'd think you'd be well advised to listen to me under those circumstances.

() It doesn't have to work -- it's successful if no enemy tests it.

But what keeps them from testing it? The threat of retaliation.
That's what we have now! That means you have to make an evaluation of why SDI is a better thing to do given all of the other options if you say SDI is the way to go.

() If it causes our enemies to spend a lot of time and resources to match it, then the diversion of their resources from their people can de-stabilize the government through the rise of dissent and unrest.

Maybe this is good, and maybe this isn't. A time-honored way of rallying the people behind you in time of internal crisis is to provoke a war. Do you really want to push the Soviets into that kind of corner?

...Is a program with a known and
predictable error rate of one wrong answer in 10,000 executions useless?

It depends on what you use the program for and how often you run it. For some things, a 1/10,000 chance of failure is quite acceptable. For others, it is quite intolerable. It depends on what depends on that wrong answer.

Herb Lin

Complex issues, complex answers

"143C::ESTELL" <estell%143c.decnet@nwc-143b.ARPA>
3 Jul 86 11:14:00 PST

There is a risk - however small - that we, like the machines we use, can begin thinking in "ones and zeros" so that everything is either "true" or "false." I believe that much of the power of computers comes from the aggregation of those "on" and "off" states to represent complex variables, text files, program logic, etc. Further, it helps to recognize sometimes that a third value of even a "logical" variable is "not initialized."

I greatly appreciate Harlan Mills' words that a good decision will come of the collective wisdom of our 535 Congressmen; they will of course be influ- enced by literally thousands of citizens(*), hopefully including many with expert technical qualifications. Moreover, I see the "official" policy at any moment as being only one "delta" of a long vector, subject to "mid course correction."

[* Note: On the other hand, congress seems heavily influenced by one citizen in particular. PGN]

Thus I assert MY OPINION that SDI should not equate to ICBM defense, even while acknowledging The President's original definition. Mr. Reagan also promised to balance the budget, in his 1980 campaign speeches. That goal has proved elusive - if not "illusive." The nation pursues updated versions of it. Similarly, President Kennedy chartered the "man on the moon" project; but that did not later deter the "grand tour of the planets" which is still going on.

It follows that I agree that working "within" the system is NOT the only way; it just happens to be my way, since I am inside. I applaud efforts of others to work outside the system, but not against it destructively. As for "opportunity lost" costs, they are always hard to measure; but we must attempt that, because it's vital. What else can we do with the SDI billions? Find the cure to the common cold? explore Mars? cut crime in half? teach Johnny to read? reduce the deficit? ALL good options. But I think we can't expect those alternatives until after '89. In the interim if we can begin a DEFENSIVE system that can be shared with allies and others as well, maybe after '90 we can re-direct many more billions towards these other worthwhile causes.

Finally, my "epsilon" in the SDI vector is to argue that the billions that DOD probably WILL spend in this decade be dedicated to concepts and objects that are feasible, and do have at least potentially useful side effects. If a major policy shift overtakes that viewpoint, I'll be very grateful. But meantime, I'd like my professional time, and my tax dollars, to go for something that I can be proud of - even after the Millennium.

Bob

[The last paragraph was a little vague and ambiguous, but if you read between the lines in this and Bob's previous messages, the intended meaning is presumably clear. However, let's all try to sharpen our thoughts and our prose on this issue in the future. And keep an eye on the computer relevancy. PGN]

Politics and Engineering Practice

Snoopy <seifert%hammer.tek.csnet@CSNET-RELAY.ARPA>
Wed, 2 Jul 86 08:51:56 PDT

In RISKS-3.13, the sad fact that politics overrules sound engineering practices is pointed out once more. Later, our fearless moderator comments on e-mail bouncing. Well, guess what? Part of the e-mail bouncing problem is political! Here at Tektronix, the mail system was suddenly changed without notice, thus either bouncing or dropping mail for days or weeks until every machine changes software, and the "new improved" addresses can be distributed throughout the world. The old addresses do not work. (Real good design there, guys!) Advance notice would have helped substantially, but politics dictated otherwise. -sigh-

Snoopy
tektronix!doghouse.GWD.TEK!snoopy (address du jour)

Multiple copies of RISKS-3.16

Kenneth Sloan <sloan@uw-tanga.arpa>
1 Jul 1986 10:16-PDT

I received (at least) two copies of RISKS-3.16. Ken Sloan

        ++
 >From NEUMANN@SRI-CSL.arpa Tue Jul 1 01:09:38 1986
 >Date: Mon 30 Jun 86 23:23:56-PDT
 >From: RISKS FORUM (Peter G. Neumann, Coordinator) <RISKS@SRI-CSL.arpa>
 >Subject: RISKS-3.16
        ++
 >From NEUMANN@CSL.SRI.COM Tue Jul 1 03:05:47 1986
 >Date: Mon 30 Jun 86 23:23:56-PDT
 >From: RISKS FORUM (Peter G. Neumann, Coordinator) <RISKS@CSL.SRI.COM>
 >Subject: RISKS-3.16
        ++

[The clue of course is the different FROM Fields. SRI-CSL went down during the wee hours of the morning in order to be reborn under its new name of CSL.SRI.COM. The mailer did its usual trick when the system bombs in the middle of a mailing -- it retries certain addresses to which it had already sent successfully. Sorry. But PLEASE NOTE THE NEW HOST NAME for RISKS and RISKS-Request: @CSL.SRI.COM. Thanks. PGN]

GTE Sprint billing problems

<Chuck.Weinstock@sei.cmu.edu [and From: Breisacher.OsbuSouth@Xerox.COM]>
2 Jul 1986 11:31-EDT

Sprint just enclosed the following notice in its latest billing:

We have recently discovered an error in our billing system related to the changeover to daylight savings time. The error may have caused some calls made in the period April 27, 1986 - May 1, 1986 to be billed incorrectly. The error has been corrected, and we are in the process of determining whether your bill was affected. If so, an appropriate adjustment, including applicable taxes and interest, will appear on a future bill...

[...although this one does not appear to have been too costly... PGN]

Search RISKS using swish-e

Report problems with the web pages to the maintainer

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 18

Tuesday, 8 July 1986

Contents

- Computer Crime in Scandinavia
  Martin Minow
- Re: Risks from inappropriate scale of energy technologies
  Henry Spencer
- Sensor technology and disinformation
  Eugene Miya
- Educating to prevent RISKS
  Steven Gutfreund
- Rash of 'Undeliverable mail'
  Chuck Price
- Info on RISKS (comp.risks)

Computer Crime in Scandinavia

Martin Minow, DECtalk Engineering ML3-1/U47 223-9922 <minow%pauper.DEC@decwrl.DEC.COM>
04-Jul-1986 0922

From the Danish newspaper, Information, (I think on 31-May-1986):

Datatapping -- The Oslo [Norway] firm, Finn Solvang A/S, has reported a Danish engineer to the police in Denmark for an attempt to get a woman employed by the firm to tap the company's computer system for valuable information on customer lists and design. The woman was offered money and instruction on how she could do the work during a weekend. The engineer is employed by a Danish firm which had collaborated with the Norwegian, but which became a competitor at the beginning of the year.

Martin Minow

(In my note on Chernobyl, I accidentally translated the Danish word "chokerade" as "choked" when it should be "shocked" -- that's what comes from writing with my fingers and not my mind. Funny that my spelling checker didn't catch it...

A few native speakers of Danish confirmed that the sentence I wasn't too certain of was reasonably translated. One said that a better translation might have been "the mathematical models used were completely wrong," making it more of a design failure than a programming bug.

Martin.)

Re: Risks from inappropriate scale of energy technologies

<decwrl!decvax!utzoo!henry@ucbvax.Berkeley.EDU>
Fri, 4 Jul 86 21:18:30 edt <RETRY OF MUCH EARLIER FAILED TRANSMISSION>

> I think that we should be pursuing a policy course which develops
> technology that can be put safely in the hands of non-technical people.
> This might take the form of small burners which use the methanol from
> organic wastes, windmills, or non-electrical solar collectors, to name a few
> possibilities. Localized, distributed technologies have many advantages,
> including ease of repair, localization of risk from outage, and major
> reductions in distribution losses and cost of distribution equipment and
> labor...

Let us not forget that distributed technologies create their own new categories of risks. The advantage of centralized resources is that much more attention can be given to keeping them safe, and they do not have to be designed to be utterly idiot-proof. (Although it helps...)

Automatic collision avoidance for airliners is imminent, while for cars it is far away. Why? Because such a system for cars would have to be cheap, easy to install and maintain, and 99.999999% reliable in a wide range of conditions despite being maintained at long, irregular intervals by largely unskilled people. Although all these characteristics certainly would be desirable for airliner systems, they are not *necessary*. Airlines can afford relatively expensive systems needing frequent attention, and can ensure that they are given regular checkouts by skilled personnel. An airliner system can also assume that a qualified pilot, prepared for the possibility of mechanical failure, is minding the store at all times. (Such assumptions are not invariably true even for airliners; the point is that they are seldom or never true for cars.)

Even disregarding this specific example, a quick look at accident rates for car travel and air travel yields interesting results for the "distributed is better" theory. Does anyone seriously believe that the level of safety attention routinely given to aircraft could possibly be given to cars?

Don't forget to compute the accident potential of distributed technologies. Methane is an explosion hazard, as witness the safety considerations for virtually any appliance using natural gas (natural gas is essentially straight methane). Windmills and solar-heat collectors don't have that problem, at least, but they do require maintenance and they are generally far enough off the ground to present a risk of accidental falls. (Last I heard, falls were the #2 [after auto accidents] cause of accidental death.) One can argue about whether lots of little accidents are preferable to a few big ones, but dead is dead either way if you're one of the victims. And it's not clear that the overall death rates are lower for distributed systems.

There is also the question of voluntarily-assumed risks versus ones one cannot avoid, but it seems to me that this case doesn't really present much of a dichotomy. If nobody builds central power plants, I really have little choice about whether I assume the risks of generating my own power. Yes, I can avoid them at the cost of major inconvenience (doing without), but I could also avoid most of the risks of centralized power at the cost of major inconvenience (move to Fiji).

Henry Spencer @ U of Toronto Zoology
{allegra,ihnp4,decvax,pyramid}!utzoo!henry

Sensor technology and disinformation

Eugene miya <eugene@ames-aurora.arpa>
7 Jul 1986 1519-PDT (Monday)

As the person who started the SDI sensor technology question which has had a couple of follow ons to Arms-d, permit me to make one comment and raise one question which Charlie Crummer@aerospace only alludes.

First, IR technology despite advances in sensor technology cannot get around the "3-body, hidden object" problem. Given a sensor and a target, if an intervening "warmer object" passes in between, the target disappears. This is an azimuth ambiguity. It sound trivial, but it is not, especially when the intervening object might be air (which does have temperature), or a mist, or other non-massive-solid. My intent is only to point this out, not some IR remote sensing.

Second, the Administration has stated a policy of disinformation with regard to SDI and letters denouncing such have appeared in Aviation Week. My question is: if we as scientists announce something as "disinformation" as one of Charlie's comments, what are all of the consequences? I can think of several including counter-announcements, the usual financial thumbscrews to funding agencies, Ellsberg type operations, and so forth. Problem is this is not a leak of information, and it's not clear to me that the SDIO can persecute this like espionage cases. Is Charlie irresponsible for revealing disinformation? Are we as scientists expected to maintain disinformation? Also, disinformation in the past has been known to backfire (another risk?).

Again the usual disclaimer that these are the opinions of the individual and not my employer, and STAR WARS is a trademark of Lucasfilm, Ltd. despite what courts say.

--eugene miya
NASA Ames Research Center
eugene@ames-aurora.ARPA

Educating to prevent RISKS

"Steven H. Gutfreund" <GUTFREUND%cs.umass.edu@CSNET-RELAY.ARPA>
Mon, 7 Jul 86 12:32 EST

RE: Jan Lee (RISKS V3 N17) on the risks of not educating casual programmers.

Your problem (in a nutshell) seem to be with the administration which needs to be made aware (educated) about the risks of under-educated programmers, than with the students themselves.

To phrase this question in full generality:

How do I make a person aware that his course of action
contains risks which he is underplaying or not cognizant of?

Classic examples of this are:

a) Try teaching a child not to touch the hot stove.
b) Teach your young and eager disciple that you have learned (via years of painful pratical experience) that he needs to take a more cautious approach (e.g. to design of large programming problems)
c) Teach your manager (who lacks modern engineering skills) that the project plan is too risky.

Approaches to attack this include:

1) Let the kid touch the stove (or the project go down the tubes)
2) Turn the issue into a confrontation (boycott the project meetings, threaten the child with loss of priviledges, etc.)
3) Try and instill the Fear of G-D in the person (long careful explanations, dissertations, memos, etc.)

There seems to be a fundamental problem in any form of directly trying to educate the unaware individual. Since what you are basically trying to do is increase the persons level of anxiety, fear, or distrust of his own thought processes. Since these emotions are not normally identified with more "rational" attitudes, there is bound to be distrust of your motives. As long as you proceed with any of the above mentioned "direct" approaches, he is bound to be AWARE of your efforts, and draw the negative conclusions.

It seems to me then that only indirect and subtle approaches will succeed.

This conclusion should be seen as especially relevent to RISKS contributors since most of them seem to be involved in publicizing fears and anxieties.

- Steven Gutfreund

Rash of 'Undeliverable mail'

Chuck Price <price@src.DEC.COM>
Tue, 8 Jul 86 11:20:05 pdt

Help! Ever since you published "License Plate Risks" in the Risks Forum,
I have been receiving a number of 'undeliverable mail' messages. A sample
is attached.

Is there any way we can stop this? I'm starting to feel like Robert Barbour.

-chuck

------- Forwarded Message [...]

Date: 8 Jul 1986 12:30:26-EDT
From: netmailer%MIT-CCC@mit-mc
Subject: Undeliverable mail
Apparently-To: <price@SRC.DEC.COM>

-- Your letter to `ghuber@MIT-MARIE' is being returned because: --

Mail undeliverable for too long

-- Returned letter follows: --

Date: 30 Jun 1986 12:32:31-EDT
From: price@SRC.DEC.COM@MIT-CCC
Date: Monday, 23 June 1986 12:56-EDT
To: RISKS-LIST:@XX.LCS.MIT.EDU, RISKS@SRI-CSL.ARPA
Subject: License Plate Risks
ReSent-From: LENOIL@XX.LCS.MIT.EDU
ReSent-To: info-cobol@ccc
ReSent-Date: Mon 30 Jun 1986 01:50-EDT

[Chuck's original message followed. This could be another risk
of undigestification. If I simply remailed individually all of the
messages in each issue of RISKS, then EACH contributor would have to
put up with the enormous number of BARF message that your moderator
otherwise puts up with! PGN]

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 19

Thursday, 10 July 1986

Contents

- Computer Literacy
  - Rick Smith
  - Bob Estell
  - Col. G. L. Sicherman
  - PGN
- Info on RISKS (comp.risks)

Computer Literacy and BASIC

<smith%umn.csnet@CSNET-RELAY.ARPA>
09 Jul 86 10:38:27 CDT (Wed)

No doubt JAN Lee's colleagues in other departments think that the literacy course is simply propaganda to improve the image of computer science and programming as serious (and difficult) work. It's a pity that it can be so easy to get a program to APPEAR to work and that most people are satisfied with apparent success. After all, a screwdriver almost looks like a chisel, and it does almost as good of a job, at least for a while.

I think Weinberg had an anecdote in "Psychology of Computer Programming" about how some DP types tried to show their managers how hard programming was by making them do some trivial BASIC programs. The managers had little trouble with their programs and went away convinced that programming was even easier than they thought.

There's a story that circulates around here about a BASIC program written several years ago. The program simulates household heating plants as part of a model of resource usage. It started as a Fortran program written at the research center of a large, local computer company. The company hires students for part-time work, one of which helped write the original Fortran program. Another student was hired later to re-code the program in BASIC. Since then the program has been sold to one of the gas industry associations and a copy was eventually sold to the Department of Energy. The students who worked on the program describe the style as a form of 'advanced spaghetti' and don't know whether to laugh or cry at the thought of it being used to plan national energy policy.

Rick Smith.
U. Minnesota

---

recognizing that one programming course is NOT enough

"143C::ESTELL" <estell%143c.decnet@nwc-143b.ARPA>
10 Jul 86 09:04:00 PST

Who should bear the responsibility for damage done by programming errors?
Everyone involved; e.g.

Colleges screen students for admission, give exams and require term projects for course credits, and charge tuition and fees for all that; thus colleges ultimately bear some responsibility for the credentials of their graduates. Those who over a period of time produce shoddy workers should lose their reputation, if not their accreditation.

Employers hire workers, give them tasks to do, and pay them for the work; and then make profits from the sale of those products or services; thus employers ultimately bear some responsibility for the products and services of their employees. Those who over a period of time produce shoddy products or services should lose money, or even go bankrupt.

Buyers seek products and services, and pay for them, so they ultimately bear some responsibility for their choices. Let the buyer beware.

Last but certainly not least, individuals who study, produce, and sell must certainly bear some responsibility for the products & services they offer. Recently, Nader has lobbied through laws making individual corporate exec- utives criminally liable for obviously defective products; e.g., when it can be proved that an auto maker produced and sold cars known to contain safety faults that led to accidental failures, injury, etc., then the man who gave the order to proceed can not hide behind a corporate mask ; a corporate fine is not enough; the man may end up in jail.

I would suggest then that when John Doe, a graduate of College of Somewhere, working for the Acme Corp., writes code that causes damage, his Alma Mater, the Acme Corp, John himself, and the "buyer" are jointly responsible.

Because buyers can't know enough to intellengly "beware" it will be often necessary to "buy insurance" in some form; that's why most of us go to MD's that are licensed by the state, and colleges that are accredited by peer groups; and why so many computing consultants "recommend IBM."

When unschooled folks set themselves up as private consultants, and hard- sell their products or services, they bear 99% of the total responsibility for the results. That might have the effect of reducing the number of freelance consultants, who charge lots of money for buzz-wordy reports. I would view that as a step forward in our industry. The good ones would not only survive, they would prosper - and be easier to find.

Finally, how can a professor convince the dean that one programming course is not enough? We can start by telling folks that since "IBM can teach you to program in FORTRAN in three days" it does NOT follow that one so trained can DO real problems in any language. By analogy, the Acme Driving School may teach one to drive in three days; that does not entitle him to a special license as a chauffeur, or to drive a 5-axle rig; and certainly does not qualify him to race a Le Mans, or Indy. Maybe if we [computer folks] turn the problem around, the others can see it better; e.g., we might suggest that our computer graduates need to appreicate physics or economics, so that they can write code that will darn near dominate the future work of physicists and economists; thus we suggest that those other departments devise one 3-hour [4-hour?] course to teach them all they need to know. After the initial [angry] retort, maybe we can enter a dialogue.

Bob

P.s. The foregoing are personal opinions, not those of my employer.

Re: RISKS-3.17 (JAN Lee on Computer Literacy)

"Col. G. L. Sicherman" <colonel%buffalo.csnet@CSNET-RELAY.ARPA>
Wed, 9 Jul 86 12:57:17 EDT

> NOW FOR MY QUESTION: How little can we get away with in preparing students
> to use the computer for problem solving and not put their eventual clients
> at risk?

JAN Lee's concern is misplaced. The "top-down" approach to teaching _about_ computers is overemphasized, perhaps because the phrase "computer literacy" sounds meaningful to educators. But the one absolute requisite for becoming a good programmer is to write programs, programs, and more programs--in any language, on any equipment available, in any environment.

I've taught hundreds of C.S. students here. By the time we graduate them, I know which students are likely to succeed: it's those who are self-motivated. The students who are just "getting an education" write no more programs than they need to, develop very slowly, and go on to write some very bad code for their employers. The students who _like_ to program write plenty of programs, learn from experience what the others try to learn by attending lectures, find alternative computers to work on or buy P.C.s if the school computer is unusable, and tend to excel in all kinds of C.S. courses.

In short, while BASIC is obviously "riskier" than Pascal, I regard the language issue as a minor one. The earlier a student starts turning problems into programs, the safer her eventual clients will be. It's futile and counterproductive to refuse to teach "just programming" on the grounds that computers are dangerous when they go wrong. Cars are dangerous, but we don't require auto mechanics to know about the thermo-dynamics of combustion engines or the social consequences of motor travel. We ask only that they be competent mechanics.

Computer Literacy (Programming versus software engineering)

Peter G. Neumann <Neumann@CSL.SRI.COM>
Thu 10 Jul 86 14:56:07-PDT

With regard to the previous message, I am in Washington this week for a conference on ensuring that a system really does what it is supposed to (COMPASS 86) and a workshop on testing, formal verification, and software engineering. This prompts me to make all sorts of comments on this issue, although they may have to wait until later.

THERE IS AN ENORMOUS DIFFERENCE BETWEEN WRITING CODE AND WRITING GOOD SOFTWARE. Any damned fool can write code. It takes a particularly perverse damn fool to write software that can be trusted to live up to rigorous requirements (which might include rugged and forgiving interfaces, reliability, maintainability, understandability, reusability, security, human safety, and so on). It also takes a lot of discipline, good taste, an instinct for elegance, training, and experience. An appropriate programming language might also help (but does not substitute for the above), as might a software development methodology -- if large and complex software is to be developed. The grave danger of computer literacy courses is indeed that they tend to endow BASIC or LISP or FORTRAN or C (or even Ada!) with magic properties. BEWARE OF SIMPLISTIC SOLUTIONS.

Writing hundreds of BASIC programs won't teach you very much about good programming style. In fact, if you did write hundreds of BASIC programs, one might suspect you hadn't learned the most important things at all -- which might even include the lesson of learning to look for a better programming language!

Allegedly "competent mechanics" have cost me hours of anguish, many dollars, and a few grave personal risks. I prefer really good, experienced mechanics who work well because they know what they are doing. If you give one an engine he has never seen before, he has to go through a learning curve -- although he will undoubtably learn much faster than the mere competent. But, the analogy is awkward -- you are asking your mechanic to keep your car working safely, not to design it from scratch in the first place.

PGN

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 20

Tuesday, 15 July 1986

Contents

- Risks of computer incompetence
  Dave Benson
- RE: educating about RISKS
  Don Lindsay
- Computer Literacy (RISKS-3.19)
  Ron Morgan
- ... and Basic
  Martin Minow
  Andrew Klossner
  PGN
- Dial-up computing
  Sterling Bjorndahl
- Research programs that pay for themselves
  Clayton Cramer
- Info on RISKS (comp.risks)

Risks of computer incompetence

Dave Benson <benson%wsu.csnet@CSNET-RELAY.ARPA>
Fri, 11 Jul 86 12:43:42 pdt

Regarding who is responsible for computer mistakes: The individual or organization who sold/licensed the software is (or should be) responsible in the eyes of the law.

In civil engineering, and a few other engineering disciplines, this responsiblity is dealt with explicitly by professional licensing by the state governments. Unfortunately, there is no professional registration for software engineers in any state. (If you know of one, please do let me know.)

The registration as a professional engineer has the same sort of effect as the licensing to practice medicine--a public statement of at least a minimal competence and a certain small amount of protection in case mistakes are made.

Not much is going to improve until the citizens agree that such a licensing procedure is necessary and software purchasers are willing to pay the extra cost this will cause--in essense, that they are willing to pay extra for lower risk.

JAN Lee and other educators might take the tack that the first course in computing is the beginning of a professional degree program. One course does not establish competence in any other field.

However, just as I need not have a professional registration in civil engineering to design and build a shed in my backyard, so I need only a little "computer literacy" to write a large range of truly useful, small-scale software. Since the results are not that remote from immediate experience, there is little risk. For example, a small program which makes pie charts can have the output quickly checked for accuracy.

A far greater concern is that most of the B-school BASIC hackers do not understand the mathematics underlying the calculations made in their small economic prediction models. Now there is a far greater risk that the model will produce wrong results, either from software misdesign or from a failure to understand the limitations of the mathematical model.

A BA or BS from a modern American university should never be taken as a license to practice. It is a minimal certification that the graduate learned something, but no guarantee of competence. Indeed, to obtain a professional registration as an engineer requires several years of practice as "engineering associate" under the direct guidance of an experienced, registered engineer. Surely the same effect holds in Business Administration, Software Engineering. It certainly does in Medicine as the new MD is required to intern before licensure for private practice.

None of this social mechanism applies to the truly large-scale software systems used in commercial and military practice today. The means of establishing low risk for any large project (software, nuclear power reactor, SDI, etc.) are imperfectly understood. I believe that it requires the right sort of organization, a particular commitment to quality which used to be exemplified by NASA. But I certainly couldn't tell you just what the characteristics of such organizations might be beyond high morale and lots of $$.

RE: educating about RISKS

<LINDSAY@TL-20B.ARPA>
Thu 10 Jul 86 12:09:30-EDT

Steven H. Gutfreund stated a problem:

> How do I make a person aware that his course of action
> contains risks which he is underplaying or not cognizant of?

Speaking as a parent, I believe in letting the kid touch the hot stove. (Yes, I really did.) Speaking as a software engineer, I believe that humor is the only effective way to communicate anxieties to students.

There are several reasons why storytelling works. For one, it sugar-coats the lesson. It makes the point more memorable. It creates the (lesser) anxiety of becoming the butt of peer amusement. And, for some students, it seems to be the only way to give them any appreciation of why they they should change their ways.

Don Lindsay

Computer Literacy (RISKS-3.19)

Ron Morgan <osmigo1@ngp.UTEXAS.EDU>
Mon, 14 Jul 86 23:46:14 cdt

As a certified all-level teacher, I'd like to say a word or two about the current "computer literacy" craze. First of all, there seems to be this constant desire to equate "computer literacy" with "programming," which ignores the fact that probably 90% of the people who use computers are *NOT* programmers. Programming is a profession, just like welding or accounting or dentistry. Courses in programming are by their very nature pre-vocational courses, regardless of whether or not they are intended as such.

Don't get me wrong; I'm not against courses in programming. A semester of it should be required of all secondary students, to give them an idea of what makes a computer tick, as well as giving them an awareness of what a proper program (stylewise) is; hopefully, they will become good software critics, at least. Students that feel an interest in becoming professional programmers should be all means have access to advanced courses that teach good style, preferably in a structure-sensitive language like Pascal. It would be a waste not to do so, in light of some of the young geniuses we are seeing more and more often these days. I know of more than one "high school hacker" that has written his or her own "bulletin board" program in *self-taught* assembly language, on such machines as the TI 99/4A and Atari 800. Recently, I talked with a 16-year-old boy that wrote a program linking two IIe's for use in running a bulletin board as a *dual-CPU* system. Sure, give these kids what they want. I'm all for it.

However, for the average Jack and Jill student, the emphasis, in my opinion, should be on developing a wide range of solid skills in USING computers. That's basically what "computer literacy" is supposed to be preparing them for, right? A society that USES computers, not a "society of programmers." I say give them courses in *real* word-processing, setting up spreadsheets, integrated applications, graphics design, telecommunications, music synthesis, database management, printer codes, statistics programs, and so on. Such knowledge, for the average student, would be far more useful, both vocationally and personally, than ten tons of required programming courses.

Ron Morgan

osmigo1, UTexas Computation Center, Austin, Texas 78712
ARPA: osmigo1@ngp.UTEXAS.EDU

UUCP: ihnp4!ut-ngp!osmigo1 allegra!ut-ngp!osmigo1 gatech!ut-ngp!osmigo1
seismo!ut-sally!ut-ngp!osmigo1 harvard!ut-sally!ut-ngp!osmigo1

Basic (a flame)

Martin Minow, DECtalk Engineering ML3-1/U47 223-9922 <minow%pauper.DEC@decwrl.DEC.COM>
11-Jul-1986 2112

I -- and a number of my friends and collegues -- have written large numbers of high quality Basic programs. These programs have been reliable, suitable to their tasks, maintainable, and efficient.

Thirteen years ago, I published a paper on writing "professional" programs in Basic (Decus European Symposium, London 1973). Very little of what I said there was particularly original: it is the sort of stuff I was taught when I learned to program way back when.

Basic has the great advantage of being easy to learn. The concepts of arithmetic and control flow seem quite natural, in many ways simpler than "structured" languages such as the descendents of Algol 58.

More importantly, Basic (Dec's RSTS/E Basic-Plus) was the first language I worked with to offer immediate feedback for syntax errors and easy incremental development. I dearly wish the people who demean Basic would invent a tool which suits their tastes, but retains the simplicity and user-friendliness of Basic.

[...] Come to think of it, it might be interesting for the Risks subscribers to compare the relative risks-to-society of a simple, intuitive langauge such as Basic against the more elegant, but harder to use, language such as ADA (or even Pascal).

Martin Minow.

Re: RISKS-3.19

Andrew Klossner <andrew%lemming.gwd.tek.csnet@CSNET-RELAY.ARPA>
Fri, 11 Jul 86 08:00:00 PDT

> "Writing hundreds of BASIC programs won't teach you very much about good programming style. In fact, if you did write hundreds of BASIC programs, one might suspect you hadn't learned the most important things at all -- which might even include the lesson of learning to look for a better programming language!"

This sort of chauvinism has no place in the RISKS forum. BASIC, like any tool, has excellent utility in its domain. For example, a complicated graphics display can be programmed easily in ANSI BASIC-86, which has a standardized statement level binding to an appropriate subset of the GKS Graphical Kernel Standard.

By now we should be beyond the point where we laugh at any language other than our favorite as being unsuitable for any serious programming endeavor.

 -=- Andrew Klossner   (decvax!tektronix!tekecs!andrew)   [UUCP]
                    (tekecs!andrew.tektronix@csnet-relay) [ARPA]

Re: RISKS-3.19

Andrew Klossner <andrew%lemming.gwd.tek.CSNET@CSNET-RELAY.ARPA>
Tue, 15 Jul 86 07:33:18 PDT

  [PGN responded to AK:
  However, the intrinsic pitfalls of BASIC are such that you might be very
  foolish to use it in a critical application. I have used several popular
  BASIC programs that can't even give reproducible results!]

You'd be hard put to come up with a commonly-used language for which this isn't true.   [Nonreproducible? Yuk. PGN]

But your original statement didn't concern itself with critical applications. You spoke of any situation in which someone had written hundreds of programs.[*] In a RSTS DP shop, popular a few years ago on PDP-11s, BASIC was the only reasonable language available, and it was quite suited to the task. In educational software development, where target systems are characterized by inexpensiveness and availability of BASIC, that language must be used if code is to be portable.
                    [* from the RISKS point of view, of course... PGN]

The point is that a knee jerk reaction that BASIC, or any single language, is inherently unsuited for any field of application smacks of elitism.

 -=- Andrew Klossner   (decvax!tektronix!tekecs!andrew)   [UUCP]
                    (tekecs!andrew.tektronix@csnet-relay) [ARPA]

Basic and critical systems

Peter G. Neumann <Neumann@CSL.SRI.COM>
Tue 15 Jul 86 21:34:28-PDT

This topic generated quite a few replies. The intent of my original comment was of course RISKS related. Certainly, a skilled and careful programmer can write excellent Basic programs, and a sloppy programmer can write bad programs in any language. But Basic has many intrinsic pitfalls that could make it harder to use in developing critical systems -- lack of modularity, abstraction and type safety, the presence of GOTOs (PLEASE let us not start that controversy again -- GOTOs are not impossible to use safely, just easier to misuse), etc. PGN

Dial-up computing

<BJORNDAS%CLARGRAD.BITNET@WISCVM.ARPA>
15 JUL 86 12:53-PST

Saturday night I dialed up to our academic computing center's VAX, as usual. Later, as I sometimes do, I absent-mindedly hit the disconnect button on my modem before logging off. "Bad form," I said, "I really shouldn't do that." But I didn't worry, because I *knew* that the network computer would log me off. It always had in the past. (I am a student at Claremont Graduate School.)

Sunday morning I dialed up again and found myself in the middle of the process I had left the night before. No login. No password. Just a '$' prompt on my screen. I had been "connected" for 13 hours. Luckily for me, no one else had tried to dial in during that time. Not even some youthful hacker with a machine to try out all the phone numbers in sequence....

Checking it out on Monday with our consultants, I found that new changes to the networking software had introduced this bug. What happened to me had also happened to several people with privileged accounts a few days earlier.

Risks to the public? My risk was basically personal. But if someone had gotten into high security accounts this way, the whole installation might have been at risk. The results of important academic research might have been lost as well. Or would that have been a benefit to the public? :-)

Sterling Bjorndahl

[We have noted previously the long-standing TENEX flaw with the similar effect -- TENEX fails to detect line loss or hangup without logout, and leaves your job logged in with its port waiting for the next person to stumble upon it. PGN]

Research programs that pay for themselves

Clayton Cramer <voder!kontron!cramer@ucbvax.Berkeley.EDU>
Wed, 9 Jul 86 17:30:54 pdt

> RISKS-LIST: RISKS-FORUM Digest, Thursday, 26 June 1986 Volume 3 : Issue 13
> Date: Thu 26 Jun 86 00:08:21-EDT
> From: Richard A. Cowan <COWAN@XX.LCS.MIT.EDU>
> Subject: Research programs that pay for themselves

[I have deleted the quote of Cowan's original message. The response from Clayton Cramer is probably not relevant, but if have erred by including something that subsequently deserves a rebuttal, then it seems that I should let the flavor of the rebuttal through. PGN]

It would be awfully good if people didn't feel they could throw any old nonsense (or even off-topic sense) into a moderated group. Mr. Cowan assertions are at least arguable, and many people would even consider false.

Assumption One: Crime is a result of unemployment, poor housing, and lack of facilities to keep young people entertained.

Assumption Two: Unemployment can be reduced by reducing the work week.

Assumption Three: Unemployment is a major problem.

[...] Clayton E. Cramer

[Clayton's message went on to counter each assertion, at some length. However, that seemed wholly inappropriate for RISKS readers, and thus I have deviated from my usual policy and truncated. PGN]

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 21

Tuesday, 15 July 1986

Contents

- Responsibility
  Willis Ware
- Programming languages and computer literacy
  Bob Estell
- Teaching about risks, BASIC, NASA, etc.
  Eugene Miya
- Programming Languages
  Matthew Kruk
- BBoard Lingo (Trojan viruses,...)
  Hank Burchard via Peter G. Neumann
- Info on RISKS (comp.risks)

Re: Responsibility (RISKS-3.20)

<willis@rand-unix.ARPA>
Wed, 16 Jul 86 09:24:36 PDT

1. Re Dave Benson's comments on responsibility. I suspect that professional licensing has one more attribute that he didn't mention; namely, it establishes some legal status and legal liability for the licensee. And it may therefore give injured parties standing to sue the licensee for injury and/or damages.

Also: what about the position that software is a consumer product? If that were to be established, then all of the consumer protection legal apparatus and consusmer protection groups would come into play -- and maybe do something useful.

IDEA: You know Susan Nycum; ask her to express an opinion on the issue and the various views.

2. Re Ron Morgan's views. I couldn't agree more with his lament that people equate "computer literacy" with "ability to program", or even worse with "ability to program properly and produce a well checked-out, tested,

and documented product that will meet specifications." They are NOT synonymous concepts, but they are related.

When the term [computer literacy] was first used and talked about more than 20 years in discussions here at Rand and elsewhere (notably by Paul Armer and Fred Gruenberger, names which I'll wager readers of RISKS don't even know), it meant simply some awareness and understanding of computery; e.g., how they work, what a program is all about, possibly some very low level of being able to use one or at least to stroke a keypunch successfully. YES, Virgina, it was keypunches in those days, not terminals.

Automatically of course, the professional programmer knew all about such things, and was computer literate. But the converse was not true: a computer literate did not automatically have all the qualifications, skills and experience of the professional -- or even semi-professional -- programmer.

The original intent was primarily to head off the fear that individuals -- and managers and organizations -- then seemed to have of computers. They were strange beasts, using strange technology, doing mysterious and invisible things and seemingly not subject to the usual precepts of management.

There was also a conviction even more than 20 years ago that computers would be important in society and in the world and would have a profound effect. One can find papers on the subject in the Joint Computer Conferences of the early 60s. Thus, it was argued that people should simply be acquainted with computery and be able to fit computers into their frames of reference comfortably, and accept them as commonplace mechanisms. Remember when you read this: I'm talking of the period when it was all mainframes and centralized computing shops, and the programming fraternity argued persuasively for and held sway in the closed-shop!

By analogy, it was like "automobile literacy" which is a characteristic that we all have even tho we don't repair our own cars or even know what's under the hood. In fact just that sort of argument was used to get the term established. Or again, being "language literate" doesn't mean that one can write Pulitzer material, only that he can read and write the language. A literate person may, but need not, be literary, educated and cultured.

We'd do well in the computer field to mind our definitions and semantics.

Willis H. Ware, Rand Corporation, Santa Monica, CA, willis @ rand-unix

---

Programming languages and computer literacy

"143C::ESTELL" <estell%143c.decnet@nwc-143b.ARPA>
16 Jul 86 08:39:00 PST

I'm somewhat repeating some of PGN's words of wisdom here, but I must

share a gem I got years ago from Lawrence Flon, then at CMU; it's
Flon's axiom, and it goes like this:

"There does not now, nor will there ever, exist a programming language
in which it is the least bit hard to write bad programs."

The entire article is worth digging out and reading; it's in SIGPLAN
Notices, October '75.

Sure, BASIC, FORTRAN, COBOL, et al, and certainly Pascal and the other
structured languages have taken us away from many of the syntactic errors
that bedeviled assembly code; like erasing the (primitive) operating
system, or jumping into a data field and crashing the processor on an
illegal op code. Cross reference checks, type checking, et al may get
us a bit away from semantic errors as well.

But what's to keep us from writing just plain wrong formulas?
This is parallel to trying to educate surgeons to prevent "bad" operations.

Switching analogies, maybe computer literacy should be the equivalent of
the "driver's license" (my earlier analogy); and programming licenses
should be the equivalent of the auto mechanic's license.

Bob

Teaching about risks, BASIC, NASA, etc. (RISKS-3.20)

Eugene Miya <eugene@AMES-NAS.ARPA>
16 Jul 1986 1008-PDT (Wednesday)

I will keep PGN's comments on Cramer's message about irrelevance in mind,
but I will come close to straying.

In his most recent RISK posting, Dr. Benson alluded to "Certification" a
time honored topic of the 1960s (certainly predating me). While I am
familiar with the so call Certified Data Processing certificate, I have met
very few CDPors. Is what these people inadequate for RISKy systems? Should
we update the CDP to include more than business type DP or should we have a
more professional (i.e., doctors and lawyers) bonding? I don't know, but it
seem a weak basis already exists and is not used, or is used weakly.

Several writers alluded to BASIC. Several more said it was irrelevant, I
agree. Real-time BASIC is (or was) used by the Navy in ship-board air
defense (I was told), and I know it is used in the Deep Space Network for
communications with unmmanned planetary probes. Those pictures you see of
controllers at the Manned Space Center are NOT programmers, they are
physicists, EEs, and other types of engineers. It was recently emphasized
in one computer ad that most did not know how to program and that some were
learning another Real-time BASIC. For them, the interaction is important;
if anything, someone needs to develop a better interactive language to
combat the problems mentioned in earlier postings. The "compiled batch"
oriented nature of most programming languages are not always conducive to

complex control systems. I am not, BTW, and have not worked in complex real-time flight systems. I do not want to worry about those RISKs, and we have all heard the stories about people with broken homes, etc. (few) because they could not handle the stress associated. Most of these flight system programmers are everyday programmers (except they work with flight qualified hardware: slower, smaller memory, etc.). Think what you will of them, they don't all program in Ada yet. If you are interested in this type of work, you should be ready for Congressional investigations (I worked on a project which had one.), and people staring you in the face and asking you tough questions about schedules (isn't hindsight wonderful).

This all finally focuses on our attitudes on how we teach risks and computing. Attitude is very important. In private correspondence to our editor, I noted am example of attitude shift in my avocation. Prior to 1946, rock climbers had a saying "The Leader (guy going first) must NOT fall." After the publication of a book entitled Belaying the Leader, climbers took a new implicit approach to belay: "The Leader will fall, what are you going to do about?" The training emphasis shifted to practicing for worse case situations. Climbing in the world went to greater levels of difficulty, fewer trained climbers were killed, and American climbing became a new standard in the world. I think we in computers are in the earliest stages of this. We don't have all the tools for a transition of ideas. But, I hope this qualitative analogy helps.

Hope I didn't stray too far.

--eugene miya
  NASA Ames Research Center

Programming Languages

<Matthew_Kruk%UBC.MAILNET@MIT-MULTICS.ARPA>
Wed, 16 Jul 86 10:11:50 PDT

You have my vote: the less intrinsic pitfalls in a programming language, the better. The "Roman Language Empire" is still young and we should strive for progressive language development or fall.

I do not deny that many "good" programs, programmers and languages exist but the day we become complacent and cease laughing is the day we become prey to our own pitfalls. We should come to expect better and not merely be satisfied.

(I do not want to start or see a "which programming language is best" debate. In some cases, the simple answer is "that which an individual is most competent at". This can be resolved in your own mind; typically, and sadly, it is resolved by your employer.)

BBoard Lingo

Peter G. Neumann <Neumann@CSL.SRI.COM>
Wed 16 Jul 86 21:18:22-PDT

From the Weekend section of the Washington Post, Friday, 11 July 1986,
on a page by Hank Burchard (Weekend at Home) devoted to home computing:

Blitz Course in Bulletin Board Lingo [Excerpts]

ARCHIVE -- Archiving is a method of compressing programs to have their original size, which makes them much faster to transmit on a modem. Since nearly everything in BBS program files is archived, an ARC(hive) coding/ decoding program is one of the first things ou should look for when cruising bulletin boards; "download" a copy for your own use. Don't use any AC program with a number higher than 5.12. AC5.13 and AC5.14 have been reported to be system-sabotaging Virus and Trojan programs.

SOFTWARE SUCKER -- The bane of sysops [SYStem OPerators]. Suckers are people who sign on to a board for one reason: to copy programs. They will download any program they find, whether or not they have any use for it, meanwhile tying up the line.

TROJAN HORSE -- One way to crash a BBS. A Trojan Horse is an innocuous-looking origran that when run reformats your harddrive, destroying all your files. To protect yourself against this, ask the store where you bought your computer, or an experienced computing friend, for a Trojan detector program. One very good one is called "Check4Bomb". Put every program you download through this before you run it. This won't catch every bad program (the inventors tend to be ingenious) but it will stop most of them.

VIRUS -- A virus program is a relative of a Trojan horse, but is usually inserted in a proven program. Users are often less suspicious of well-known programs.

[I toss this one in for good measure. The RISKS OF BBOARDS are rampant, but so are the RISKS OF OVERSIMPLIFICATION. PGN]

Search RISKS using swish-e

Report problems with the web pages to the maintainer

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 22

Saturday, 19 July 1986

Contents

- Nostalgia
  Mike Williams
- Flames about BASIC
  Jim Anderson
- More on risks of teaching "just" programming
  Herb Lin
- Responsibility for Computer Actions
  George S. Cole
- CDP and Certification
  Andy Glew
- The undetected hang-up risk (more)
  Ted Lee
- Info on RISKS (comp.risks)

Nostalgia

"John Michael (Mike) Williams" <JWilliams@DOCKMASTER.ARPA>
Thu, 17 Jul 86 17:34 EDT

Willis' comment in connection with computer literacy in the old days:

>Remember when you read this: I'm talking of the period when
>it was all mainframes and centralized computing shops, and the programming
>fraternity argued persuasively for and held sway in the closed-shop!

triggered all sorts of memories I'm sure Willis shares.

Surely we both remember the Bendix G-15? the Monrobot? the CDC 160A (that motherless PPU)? Yes, there were big IBM 704 and UNIVAC II shops, but there were also IBM 604 Punched Card Electronic Calculators, and UNIVAC 40/60/120s; the latter I remember being used for critical airframe and weapons system calculations by then-Douglas Aircraft in 1956, when I joined the industry.

I don't remember, if I ever knew, what computers were used to support the

Comet and Electra I designs, but perhaps there may be a connection between their sorry record and RISKS. In any case, the problem of distributed small computing environments has always been with us, if on a smaller scale.

Mike Williams, System Development Corp. McLean VA

Flames about BASIC

<JPAnderson@DOCKMASTER.ARPA>
Thu, 17 Jul 86 22:45 EDT

Those of your readership who bristle when one programming language or another is put down for any reason might like to read what has to rank as the ultimate rebuttal. I refer of course to Howard E. Tompkins paper "In Defense of Teaching Structured COBOL as Computer Science (or Notes on being Sage Struck). It appeared in SIGPLAN notices, V18,4 of April 1983. A real hoot!

Jim

More on risks of teaching "just" programming

<LIN@XX.LCS.MIT.EDU>
Sat, 19 Jul 1986 02:23 EDT

My own feeling is that for for "computer literacy" in the general populace (rather than say for engineers or economists who will have to write programs), programming is mostly irrelevant. The most important notions for everyone to have (that is after all the meaning of "literacy") are those related to procedures: what procedures are, what input is, what output is, how input can be related to output and so on. Being able to ask the question "But how can the computer know to do X?" in a meaningful way, and puzzling out the answer to that question is in my view a whole lot more important than knowing the syntax of PASCAL or BASIC.

The problem is ultimately related to clear thinking, and how to teach people to do THAT.

[We have included various somewhat redundant responses on this topic in recent RISKS, because the points being made are IMPORTANT but OFTEN IGNORED. There is no substitute for style, elegance, care, and -- above all -- understanding what you are doing. PGN]

Responsibility for Computer Actions

<cole.pa@Xerox.COM>
31 Dec 00 16:30 PST

Responsibility for a computer foul-up can realistically be laid anywhere from the individual operator's feet (for placing the wrong hard disk in or plugging in the wrong power supply) to the hardware designer's feet (for allowing ungrounded power plug-ins) to the system programmer's feet to the compiler design team's feet (group shot) to the application's designers' feet (another group shot?) ... all depending on what is the "source" of the failure.

Presuming for the nonce that the fault lies in the application software (not in its implementation, via the transition from high-level to machine code or transition from electronic state to electronic state), there still remains a problem of determining who is responsible. Who provided the algorithm? The implementation? The specification? Did anybody perform a mathematical theorem validation? Could such realistically be done for the entire program? (Hah.) Hindsight allows a (relatively) easy post-mortem that shows "this step" could have been validated (and thus had the error shown up), often enough. But the program is a SYSTEM, and the safeguards are at this point far from perfect.

Ought they be perfect? Think how much that would cost.

Rather than tacking terms like "responsibility" to the entire spectrum of computer programs, it would make more sense (legally and ethically) to designate the principles and requirements for liability to be attached for an injury, and let the moralists be concerned with the responsibility. (Responsibility can NEVER be attached, no matter how hard it is thrown; it is only accepted. But I would far rather have people programming with or for me who voluntarily accept responsibility, since they then provide the best protection.)

Professional licensing, which requires the establishment of minimal standards, allows actions based on malpractice to be brought. As long as this licensing is voluntary and not mandatory the market can help establish responsibility -- for then the product seller who hires an unlicensed programmer to produce the core program will have to consider whether they might be charged with negligence.

Standard applications, however, should only be subject to strict "products" liability where there is a standard operating environment. If a program specifies that it is designed to operate on an Apple II-E with an Epson MX-80 or FX-80 printer, (or some set of CPU chips, terminals, and printers with a set of standard operating systems), any user who goes to a different environment (even if somebody else promised it would be identical, or just compatible) has no one but himself to curse. The difference between a hammer and a consumer computer application is (realistically) indifferent in terms of consumer law -- if you use a hammer as a wedge or a support for some scaffolding, you can hardly cry foul when it fails at a task for which it is not designed.

(Of course, the above is complicated by some rulings that "foreseeable misuses" allow liability. The consumer applications computer company will want to restrict the range by specifying where it guarantees its product , and will want to extend the probable hardiness to a penumbra of likely modifications beyond that to prevent mishaps.)

George S. Cole, Esq.

CDP and Certification

Andy "Krazy" Glew <aglew%ccvaxa@gswd-vms.ARPA>
Thu, 17 Jul 86 09:11:19 cdt

Eugene Miya asks whether the CDP is a level of professional certification. I do not have a CDP, but I passed the Certified Computer Programmer (CCP) exam in Systems Programming which is also given by the Institute for the Certification of Computer Professionals (ICCP).

Does passing the exam itself indicate any level of competence? No - I would expect first year engineering students to be able to pass it with no difficulty. However, the fact that someone is serious enough about `professionalism' to go out and get certified probably indicates something about his character, if not his abilities. Obviously, the certification process must become more stringent - the new requirement for periodic recertification is a step in the right direction.

A secondary effect of `professional' certification is that you are expected to subscribe to a code of ethics. Many people deride these, but I know that I, at least, have them in the back of my mind when I consider systems whose failure can harm people. `Empty symbology' has a powerful psychological effect: wearing an Iron Ring reminds me about an oath I took with much rattling of chains that I would never "pass bad workmanship". The ancient Greeks used to pour libations to gods they knew weren't there.

Why take something like the CCP? For frankly mercenary reasons - I took it to increase my chances of getting a job. But also because I am familiar with the history of engineering as a profession in Canada and Great Britain (engineering isn't a profession in the United States yet, is it?) and though that the ICCP might be the beginning of something similar for software engineering / computer science / programming.

What would distinguish such a profession from the present situation? Purely and simply, liability. A professional is liable for his actions, not just to the best of his ability, but to the limits of knowledge in his field.

Liability is a great incentive for taking proper care of your work. To the extent that care, the highest reasonable level of care that we can expect humans to provide, can reduce the chance of failure in software systems, professionalism is a good thing.

Andy "Krazy" Glew. Gould CSD-Urbana.   USEnet: ihnp4!uiucdcs!ccvaxa!aglew
1101 E. University, Urbana, IL 61801   ARPAnet: aglew@gswd-vms

---

The undetected hang-up risk (more)

<TMPLee@DOCKMASTER.ARPA>
Fri, 18 Jul 86 03:08 EDT

When our local GTE Telenet office finally installed its 2400 baud service I discovered the same problem referred to in the penultimate Risks: if the line dropped, there was a very good chance the local Telenet machine did not detect it and one could later dial back in. Several times i dialed in and

found myself in the middle of someone else's connection; I also, of course,
after several hours (almost a day one time, I seem to remember) was able to
dial back in and find myself connected to my original host system. It took
several weeks of trouble reports, as well as calls from "high government
officials" (the computer I was using was this one: the folks at the
National Computer Security Center were not, as one would hope and expect,
pleased) before Telenet acknowledged there was a problem and did something
about it. I seem to remember that it was simply an ill modem, but the
experience was enlightening.

Ted

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 23

Tuesday, 22 July 1986

Contents

- Re: Comet and Electra
  Jim Horning
- 100,000 Late Phone Bills
  Mike McLaughlin
- Types of "Programming"
  Henry Schaffer
- Info on RISKS (comp.risks)

Re: Comet and Electra

Jim Horning <horning@src.DEC.COM>
Mon, 21 Jul 86 14:19:04 pdt

The message from Mike Williams (RISKS-3.22) reminded me of two stories that have been passed down through the oral tradition. I have no direct evidence concerning either. Perhaps some readers of Risks have evidence that could help set the record straight?

- A numerical analyst once explained to me why all modern airliner windows have rounded corners: Anyone concerned with solving partial differential equations knows that square corners lead to singularities. He said that the Comet crashes were traced to metal fatigue at the (square) corners of its windows. (He concluded that airplane designers should study Numerical Analysis.) Does anyone know whether computers were used in the design of the Comet?

- I also heard that the structural defect in the Electra I wing design had not been caught by the stress analysis program because of an undetected overflow on a critical run. Can anyone provide documentation for this? (I think this story was on the grapevine at the NATO Software Engineering Conferences in 68-69.)

These pieces of our mythology are worth documenting or debunking. There may be valuable lessons to be learned from them, and we ought not to insist on learning them the hard way.

Jim H.

100,000 Late Phone Bills

Mike McLaughlin <mikemcl@nrl-csr>
Mon, 21 Jul 86 16:03:50 edt

Excerpted from the Washington Post, Saturday, 9 July 1986, page D1.
[Omissions... (bridges) and [comments] as shown.]

More Than 100,000 Getting Months-Late (telephone) Bills
By Nell Henderson, Washington Post Staff Writer

More than 100,000 Chesapeake & Potomac Telephone Co. customers might think they've had a summer vacation from telephone bills.
But yesterday the company said the vacation is over: The bills are on the way after a two-month delay.
The customers... have not received bills for local or long-distance service or both - since a computer tape failure in mid-May.

The high-tech roots of the problem were "flaws" in computer tapes that were programmed for preparing the bills, (a spokesman said). "The problem erases itself," he added.
The low-tech solution was to use people to put the billing information into the system, using separate records of the calls, he said.
The result was that many of the customers did not receive phone bills for several months....

(A) customer... was told to call... if he has any trouble paying the entire bill at once.
"We would be lenient on payment, and would be glad to speak to customers on an individual basis . . . We're sorry for any inconvenience,"...
The problem also affected an unknown number of bills for long-distance service provided by MCI Communications Service...

Types of "Programming"

Henry Schaffer <ecsvax!hes%mcnc.csnet@CSNET-RELAY.ARPA>
Fri, 18 Jul 86 23:37:38 edt

"Programming" encompasses much more than the use of the traditional languages (Basic, Ada, or whatever.) Entering formulas in a spreadsheet or specifying record and report structures in a database are also programming - in higher-level, albeit specialized, languages. Thus JAN Lee *is* teaching his students to program, and in the most appropriate and productive manner. They can learn something quite important and useful in this part of the class. It is the other faculty/administrative objectors (the ones who want to have 4 weeks of traditional language put in) who are asking for something both unproductive (most of the students will neither learn new concepts nor something useful) and risky.

There is an implicit understanding about a terminal course - that you've been carried along far enough so that you can use what you've been taught. A student who finishes one semester of a CSC sequence knows that he/she is not through learning, and should not presume (one hopes) to take on responsibilty for a critical application program. However, a student who is taught that programming is 4 weeks of a survey course in computing might not be so timid! (I assume that these students will not usually take any more programming - if they generally did then there wouldn't have been the pressure to push programming into JAN Lee's course.)

Our university (NCSU) has recognized that the details of the type of "programming" needed are dependent on the discipline, and can variously include spreadsheets, statistical packages (I can argue that one can "program" in SAS), etc., and also the more traditional languages.

--henry schaffer n c state univ ...mcnc!ecsvax!hes (uucp)
tsches@ecsvax.bitnet

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 24

Thursday, 24 July 1986

Contents

- Comet and Electra
  - Jerry Saltzer
  - Marv Zelkowitz
  - Don Chiasson
  - Bard Bloom
- No gasoline because the computer is down?
  - Jim Barnes
- HBO Hacker Captain Midnight Caught
  - via Geoff Goodfellow
- Info on RISKS (comp.risks)

Re: Comet and Electra

Jerome H. Saltzer <Saltzer@ATHENA.MIT.EDU>
Tue, 22 Jul 86 23:26:59 EDT

> \- I also heard that the structural defect in the Electra I wing design had
> not been caught by the stress analysis program because of an undetected
> overflow on a critical run. Can anyone provide documentation for this? (I
> think this story was on the grapevine at the NATO Software Engineering
> Conferences in 68-69.)

In case it helps anyone recall where that one might be documented: the version of the story that came through here had it that some piece of simulation input data was typed with the wrong minus sign. (The commonly available version of the 026 key punch had a minus sign and a hyphen as distinct characters. And the input format conversion routines in those days were both unforgiving and silent about errors.)

Jerry

Re: Comet and Electra

Marvin Zelkowitz <mvz@aaron.cs.umd.edu>

Wed, 23 Jul 86 09:57:25 edt

Horning's recent comment reminds me of two related items:

- On the Electra I wing design defect: My version of the story goes that the undetected overflow error was finally detected when these "correct" programs were used as benchmarks for a new computer (a Burroughs I think), which gave radically different answers. I do not have any proof of this, but it might give some additional help in tracking it down.

- On overflow detection: In the late 60s, a certain vendor's FORTRAN did not detect overflow. At a users' group meeting, the vendor offered to add overflow detection at an execution penalty of one instruction per arithmetic operation (e.g., branch-on-overflow). This was voted down. The only conclusion is that users would rather be fast than right. The issue for RISKS is "Are these people the ones 'still in control'?"

--Marv Zelkowitz

Re: Comet and Electra

Don Chiasson <CHIASSON@DREA-XX.ARPA>
Wed 23 Jul 86 09:17:42-ADT

> From: horning@src.DEC.COM (Jim Horning)

> - A numerical analyst once explained to me why all modern airliner windows
> have rounded corners: Anyone concerned with solving partial differential
> equations knows that square corners lead to singularities. He said that the
> Comet crashes were traced to metal fatigue at the (square) corners of its
> windows. (He concluded that airplane designers should study Numerical
> Analysis.)

Most engineers know that any sharp corner on a stressed member will cause an increase of actual stress over the nominal calculated stress, and the ratio of these is called the stress concentration factor, K. The value of K is sort of inversely proportional to the radius of curvature of the discontinuity. High K is the reason cracks propagate so well. The temporary fix for a crack is to drill a hole at the end of the crack which increases the radius of the "corner" and decreases K. It is standard design practice to avoid sharp corners. Stress concentration is usually discussed in design textbooks without going into the differential equations: there are lots of tables.

This brings up a problem encountered in computer applications: the difficulty of a programmer learning the standard practices of a field in which he is working. Engineers know about stress concentration, but programmers and mathematicians may not.

> - I also heard that the structural defect in the Electra I wing design had
> not been caught [...]. Can anyone provide documentation for this?

I can't give a direct answer to this, but I know that a mid 60's computer which was heavily used in scientific and engineering applications had very poor accuracy in its trig package. Is this perhaps the same topic? (Or was the Electra designed in the 50's??) Note: I can identify the manufacturer and machine, but feel that if I did so, I would be potentially libelous.
Don Chiasson

Re: Comet and Electra

Bard Bloom <BARD@XX.LCS.MIT.EDU>
Wed 23 Jul 86 11:44:00-EDT

[Structural defect in the Electra I wing design, again. See Jerry, above.]

I don't know about this, but I was trying to move some software in Fortran from an IBM to VAX for McDonnell-Douglas one summer. The program on the VAX kept dying, with a message to the effect of "I can't take a sine of a number this large". The program was trying to take sines of large (order of 10^20) numbers in 16-digit arithmetic. The first thing that the sine routine does is reduce its argument modulo pi, which loses *all* of the precision of the 20-digit number. The VAX's software generated an error about this. The IBM did not; and the programmers hadn't realized that it might be a problem (I guess). They had been using that program, gleefully taking sines of random numbers and using them to build planes, for a decade or two.

No gasoline because the computer is down?

Jim Barnes <decvax!wanginst!infinet!barnes@seismo.CSS.GOV>
Wed, 23 Jul 86 13:56:44 edt

Last Friday, on my way home, I stopped at the local gasoline station to "fill 'er up". However, they could not pump any gas because the "computer was down". It seems that the pumps at the station were the new kind (with the digital displays for price per gallon, total, etc.) and were linked through to some computer somewhere. Who would have thought that a computer failure could prevent us from being able to purchase gasoline? But now that I think of it, all those new point of sale terminals linked to a computer could be in trouble if the computer fails.

It used to be that this kind of problem would occur only if there was an electrical power outage, but now just having the computer down can cause the same problem.

decvax!wanginst!infinet!barnes   Jim Barnes

HBO Hacker Captain Midnight Caught

23 Jul 1986 17:08-PDT

JACKSONVILLE, Fla. (AP) - Investigators using a complicated process of elimination have unmasked ''Captain Midnight,'' who admitted in court he overrode HBO's satellite delivery system to transmit a message.

John R. MacDougall, owner of a home satellite dish business in Ocala that officials said was hurt by cable companies' decisions to scramble their signals, agreed to plead guilty to illegal transmission of a satellite signal in exchange for a $5,000 fine and one year probation.

He could have faced a maximum $10,000 fine and a year imprisonment.

MacDougall, who was released on a $5,000 bond, and his attorney, John M. Green Jr., refused to comment as they left the federal court building Tuesday after entering the plea before a U.S. magistrate.

Sentencing is set for Aug. 26 and MacDougall can retract his plea if the judge will not accept the arrangement.

Early on April 27, MacDougall was the only one working at a satellite transmission center called Central Florida Teleport with the kind of equipment needed to disrupt the HBO signal, officials said.

Although the video sneak attack was only a minor annoyance to HBO and its viewers, the Federal Communications Commission launched a massive investigation because of the potential problems a less selective video hacker might cause.

''The potential for damage to critical satellite frequencies cannot be underestimated,'' said Richard M. Smith, chief of the FCC's field operations bureau. He noted that critical telephone calls, air traffic control, military data and medical information are sent by satellite and that even an accidental interruption of one of these messages could cause dire consequences.

On April 27, HBO viewers saw a message replace the movie ''The Falcon and the Snowman.'' The message said:

''Good Evening HBO

''From Captain Midnight

''$12.95 month

''No way!

''(Showtime Movie Channel beware.)''

The wording was an apparent reference to HBO's decision to scramble its satellite-delivered signal so it could not be watched by those not paying for HBO, officials said.

''His company was sustaining substantial losses because of the scrambing of HBO and threats of other scrambling,'' said Assistant U.S. Attorney Lawrence Gentile III.

MacDougall also interrupted HBO video signals on April 20, when he transmitted a color bar pattern, officials said.

On Jan. 15, HBO became the first cable TV network to scramble its signal full time. Showtime and The Movie Channel scrambled their programming full time on May 27.

The scrambling makes pictures unwatchable without a descrambler and slowed sales of satellite dishes.

Of 580 satellite facilities with a transmitting dish large enough to overpower HBO's signal, less than a dozen had sufficient power and the right kind of electronic typewriter to write the protest message Captain Midnight transmitted, investigators said.

The investigation focused on Ocala after a tipster vacationing in Florida reported to the FCC an overheard telephone call about Captain

Midnight. The tipster provided the caller's description and license plate number.

The caller who was overheard was not the suspect, but the FCC said the information provided proved extremely beneficial.

[The L.A. Times refined this a little, after noting that there were only 580 appropriate candidate facilities:

"By studying tapes of the illegal video signal, the FCC's field staff concluded that the message had been generated using a specific make and model of character-generator device to transmit symbols, such as letters and numbers, onto a television screen.

"After visiting those plants, investigators had three prime suspects, including MacDougall. When he was notified he was a suspect, MacDougall turned himself in."

This seems like a nice bit of detective work, and certainly presents an interesting risk for would-be perpetrators -- somewhat like radioactive traces in dyes, watermarks in paper, imperfections in certain characters on a typewriter or printer, and voiceprints (all of which have been used successfully to identify or subset culprits). On the other hand, the smart perpetrator, aware of such tell-tale signatures, might figure out a way to spoof someone else's tell-tale, similar to changing the answer-back drum on a teletype or hacking your cellular telephone identifier (as noted in a previous RISKS by Geoff). Will this case escalate the sophistication of satellite attacks? PGN]

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 25

Thursday, 21 July 1986

Contents

- Petroski on the Comet failures
  Alan Wexelblat
- Re: Comet and Electra
  Douglas Adams
- On the dangers of human error
  Brian Randell via Lindsay Marshall
- Software Paranoia
  Ken Laws
- Royal Wedding Risks
  Lindsay Marshall
- How to Think Creatively
  John Mackin
- Dangers of improperly protected equipment
  Kevin Belles
- Info on RISKS (comp.risks)

Petroski on the Comet failures

Alan Wexelblat <wex@mcc.com>
Thu, 24 Jul 86 12:02:41 CDT

Henry Petroski's book _To Engineer is Human_ has a segment discussing the Comet crashes and the detective work done to figure out why they occurred (pages 176-184). The story he tells makes no mention of curved or rounded window corners. The highlights:

- On May 2, 1953, a de Havilland Comet was destroyed on takeoff from Dum-Dum Airport in Calcutta, India. The Indian Government Board of Inquiry concluded officially that the accident was caused by some sort of structural failure either due to a tropical storm or to pilot overreaction to storm conditions.

- The Comet was flown "off the drawing board"; no prototypes were ever built or tested.

- On January 10, 1954, a Comet exploded after takeoff from Rome under mild weather conditions. The plane was at 27,000 feet so the debris fell into a large area of the Mediterranean. Not enough was recovered to allow any conclusion on why the crash had occurred.

- On April 8, 1954, another flight leaving Rome exploded. The pieces from this one fell into water too deep to allow recovery, so more pieces from the previous crash were sought and found.

- Investigators eventually found the tail section which provided conclusive evidence that the forward section had exploded backward. The print from a newspaper page was seared into the tail so strongly that it was still legible after months in the Mediterranean.

- The question now was WHY did the cabin explode? The reason was found only by taking an actual Comet, submerging it in a tank of water and simulating flight conditions (by pressurizing and depressurizing the cabin and by hydraulicly simulating flight stresses on the wings).

- After about 3000 simulated flights, a crack appeared at a corner of one cabin window which rapidly spead (when the cabin was pressurized) and the cabin blew apart.

- Analysis finally showed that rivet holes near the window openings in the fuselage caused excessive stress. The whole length of the window panel was replaced in the later Comet 4 with a new panel that contained special reinforcement around the window openings.

Although Petroski doesn't give his sources directly, much of his material appears to be drawn from the autobiography of Sir Geoffrey de Havilland (called _Sky Fever: The Autobiography_, published in London in 1961) and from a book called _The Tale of the Comet_ written by Derek Dempster in 1958.

In general, I recommend Petroski's book; it's quite readable and has lots of material that would be interesting to we RISKS readers. Of particular interest is the chapter called "From Slide Rule to Computer: Forgetting How it Used to be Done." It's an interesting (if superficial) treatment of some of the risks of CAD.

Alan Wexelblat
ARPA: WEX@MCC.ARPA
UUCP: {ihnp4, seismo, harvard, gatech, pyramid}!ut-sally!im4u!milano!wex

Currently recruiting for the `sod squad.'

Re: Comet and Electra

Adams Douglas <crash!pnet01!adamsd@nosc.ARPA>
Thu, 24 Jul 86 07:43:49 PDT

It was my understanding that the problem with the early Electras was whirl-mode flexing of the outboard half of the wing. I had heard that Lockheed reassigned its few then-existing computers to full-time research on the problem. But it was also my understand that the original design cycle for the Electra did not involve computer assistance at all--they weren't being used for aircraft "simulation" that early (1948?).

On the dangers of human error [contributed on behalf of Brian Randell]

"Lindsay F. Marshall" <lindsay%kelpie.newcastle.ac.uk@Cs.Ucl.AC.UK>
Thu, 24 Jul 86 11:28:28 bst

[From brian Fri Jul 18 17:30 GMT 1986]
The following article appeared in the Guardian newspaper (published in London and Manchester) for Wed. July 16. The author, Mary Midgely is, incidentally, a former lecturer of Philosophy at the University of Newcastle upon Tyne. Brian R. was pleased to see such a sensible discussion in a daily newspaper of the dangers of human error that he thought it worth passing on to the RISKS readership, so here it is.....

IDIOT PROOF

Little did I know, when I wrote my last article about human error, that the matter was about to receive so much expensive and high-powered attention. Since Chernobyl, it has been hard to turn on television without receiving more official reassurance that accidents do not happen here. Leading the chorus, the chairman of the Central Electricity Generating Board came on the air to explain that, in British nuclear reactors, human error has been programmed out entirely. Other equally impressive testimonies followed. Even on these soothing occasions, however, disturbing noises were sometimes heard. During one soporific film, an expert on such accidents observed that human error is indeed rather hard to anticipate, and told the following story.

A surprising series of faults occurred at a newly-built nuclear power station, and were finally traced to failure in the cables. On investigation, some of these proved to have corroded at an extraordinary rate, and the corroding substance turned out to be a rather unusual one, namely human urine. Evidently the workmen putting up the power-station had needed relief, and had found the convenient concrete channels in the concrete walls they were building irresistibly inviting. Telling the tale, the chap reasonably remarked that you cannot hope to anticipate this kind of thing - infinitely variable human idiocy is a fact of life, and you can only do your best to provide against the forms of it that happen already to have occurred to you.

This honest position, which excluded all possible talk of programming it out, is the one commonly held by working engineers.

They know by hard experience that if a thing can go wrong it will, and that there are always more of these things in store than anybody can possibly have thought of. (Typically, two or three small things go wrong at once, which is all that is needed). But the important thing which does not seem to have been widely realised is that hi-tech makes this situation worse, not better.

Hi-tech concentrates power. This means that a single fault, if it does occur, can be much more disastrous. This gloomy truth goes for human as well as mechanical ones. Dropping a hammer at home does not much matter; dropping it into the core of a reactor does. People have not been eliminated. They still figure everywhere - perhaps most obviously as the maintenance-crews who seem to have done the job at Chernobyl, but also as designers, sellers and buyers, repairers, operators of whatever processes are still human-handled, suppliers of materials, and administrators responsible for ordering and supervising the grand machines.

What follows? Not, of course, that we have to stop using machines, but that we have to stop deceiving ourselves about them. This self-deception is always grossest over relatively new technology. The romanticism typical of our century is altogether at its most uncontrolled over novelties. We are as besotted with new things as some civilisations are with old ones.

This is specially unfortunate about machines, because with them the gap between theory and practice is particularly stark. Only long and painful experience of actual disasters - such as we have for instance in the case of the railways - can ever begin to bridge it. Until that day, all estimates of the probability of particular failures are arbitrary guesses.

What this means is that those who put forward new technology always underestimate its costs, because they leave out this unpredictable extra load. Over nuclear power, this is bad enough, first, because its single disasters can be so vast - far vaster than Chernobyl - and second, because human carelessness has launched it before solving the problem of nuclear waste.

Nuclear weapons, however, differ from power in being things with no actual use at all. They exist, we are assured, merely as gestures. But if they went off, they would go off for real. And there have been plenty of accidents involving them. Since Chernobyl and Libya, people seem to be noticing these things. Collecting votes lately for my local poll on the Nuclear Freezen project, I was surprised how many householders said at once: "My God, yes, let's get rid of the things." This seems like sense. Could it happen here? Couldn't it? People are only people. Ooops - sorry...

Software Paranoia

Ken Laws <Laws@SRI-STRIPE.ARPA>
Thu 24 Jul 86 17:40:04-PDT

From: Bard Bloom <BARD@XX.LCS.MIT.EDU>
The VAX's software generated an error about this. The IBM did not; and the programmers hadn't realized that it might be a problem (I guess). They had been using that program, gleefully taking sines of random numbers and using them to build planes, for a decade or two.

Let's not jump to conclusions. Taking the sine of 10^20 is obviously bogus, but numbers of that magnitude usually come from (or produce) other bogus conditions. The program may well have included a test for an associated condition <>after<< taking the sine, instead of recognizing the situation <>before<< taking the sine. Poor programming practice, but not serious.

A major failing of current programming languages is that they do not force the programmer to test the validity of all input data (including returned function values) and the success of all subroutine calls. Debugging would be much easier if errors were always caught as soon as they occur. The overhead of such error checking has been unacceptable, but the new hardware is so much faster that we should consider building validity tests into the silicon. The required conditions on a return value (or the error-handling subroutine) would be specified as a parameter of every function call.

I tend to write object-oriented subroutines (in C) that return complex structures derived from user interaction or other "knowledge-based" transactions. Nearly every subroutine call must be followed by a test to make sure that the structure was indeed returned. (Testing for valid substructure is impractical, so I use NULL returns whenever a subroutine cannot construct an object that is at least minimally valid.) All these tests are a pain, and I sometimes wish I had PL/I ON conditions to hide them. Unfortunately, that's a bad solution: an intelligent program must handle error returns intelligently, and that means the programmer should be forced to consider every possible return condition and specify what to do with it.

Errors that arise within the error handlers are similarly important, but beyond my ability to even contemplate in the context of current languages.

Expert systems (e.g., production systems) often aid rapid prototyping by ignoring unexpected situations -- the rules trigger only on conditions that the programmer anticipated and knew how to handle. New rules are added whenever significant misbehavior is noticed, but there may be no attempt to handle even the full range of legal conditions intelligently -- let alone all the illegal conditions that can arise from user, database, algorithm, or hardware errors. I like expert systems, but from a Risks standpoint I have to consider them at least an order of magnitude more dangerous than Ada software.

-- Ken Laws

Royal Wedding Risks

"Lindsay F. Marshall" <lindsay%cheviot.newcastle.ac.uk@Cs.Ucl.AC.UK>
Thu, 24 Jul 86 13:46:31 gmt

Yesterday (23rd) we lost all power to our machine room when a circuit breaker blew. The cause of this was a glitch which hit us at about 13:50 P.M. This was approximately the time that the main Royal Wedding television coverage stopped...........

How to Think Creatively

<munnari!basser.oz!john@seismo.CSS.GOV>
Thu, 24 Jul 86 18:21:08 EST

Recent comments in Risks about ``computer literacy'' lead Herb Lin
to comment that:

> The problem is ultimately related to clear thinking, and how to teach
> people to do THAT.

This reminded me of some mail I received last year, from a staff member
here who was teaching a first-year course on data structures. His mail,
which was sent to a number of us here, was a plea for assistance as to
the right way to respond to some mail he had received from one of his
students. The student's mail said:

> Dear Jason,... You have really done a great job on IDS. It really helped to
> clear a lot of lingering doubts Lent term left behind. Thanks a lot
> again. Could you advise on how to think creatively. I can't "see" a
> program naturally and think deep enough to make the required alterations...

None of us really knew how to answer that.

John Mackin, Basser Department of Computer Science,
        University of Sydney, Sydney, Australia

john%basser.oz@SEISMO.CSS.GOV
{seismo,hplabs,mcvax,ukc,nttlab}!munnari!basser.oz!john

Dangers of improperly protected equipment

Kevin Belles <crash!pnet01!kevinb@nosc.ARPA>
Thu, 24 Jul 86 01:08:50 PDT

Is there any device or devices that protect not only the active lines
but the ground lines as well from surge, spike, and EMI-type disturbance?
My system appears to have been victimized, thanks to our local electric
utility, by the ground for my apartment complex being raised, which caused
damage to all the damage to all the grounded equipment on my home computer
system, save some cards apparently protected by my boat-anchor power supply,
and the fact that each card in my cage is independently regulated. In my
case, the surge entered the ground and apparently corrupted my main floppy
drive supply to the point where it propagated along the 8" and 5 1/4"
cables, destroying the logic boards on all drives and the dynamic memory,
which was being accessed at that time. It also managed to get my printer, on
another leg entirely, while miraculously missing my terminal and modem. This
completely bypassed the fuses and only a trace on the controller board being
opened saved the rest of my system being damaged. Result: 3 dead DSDD 8"
drives, 1 dead SSDD 5 1/4" drive, 3 drive power supplies, 1 dot-matrix
printer, 1 64K DRAM board, and a floppy controller board. Dollar cost:
estimated minimum of over $2000.00 if equipment is replaced by new, with no

cost for loss of access being figured in.

Let this be a warning: Protect your equipment! Any investment in anti-surge equipment, anti-spike equipment, and UPSs are investments in your computing future.

Kevin J. Belles - UUCP {sdcsvax,noscvax,ihnp4,akgua}!crash!pnet01!kevinb

(Disclaimer: Anything I may say is my opinion, and does not reflect
    the company I keep. KjB)

Search RISKS using swish-e

Report problems with the web pages to the maintainer

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 26

Saturday, 26 July 1986

Contents

- DIVAD
  Herb Lin
- Royal wedding risks -- common change modes
  Don Chiasson
- Security and dialbacks
  David I. Emery via Herb Lin
- Info on RISKS (comp.risks)

DIVAD

<LIN@XX.LCS.MIT.EDU>
Sat, 26 Jul 1986 00:39 EDT

Some time ago there was a flap about whether or not DIVAD did or did not shoot at a latrine fan. [See Doug Schuler in RISKS-3.1, with subsequent discussion in RISKS-3.3, 4, 5.] I have documentation now from a person who should know: Richard DeLauer, former Undersecretaty of Defense for Research and Engineering in the first Reagan term. He says it did, and that it was supposed to do that. See [MIT] Technology Review, July 1986, page 64.

Royal wedding risks -- common change modes

Don Chiasson <CHIASSON@DREA-XX.ARPA>
Fri 25 Jul 86 10:25:41-ADT

Phenomena like this are well known by the CEGB (Central Electricity Generating Board) engineers. Operation of a power grid assumes that the load does not change suddenly, indeed sudden changes can cause instability. Anyway, it is well known in the U.K. (I'm not sure about the U.S. and Canada) that the largest power surge is at the end of Coronation Street, or one of the other soaps, when everyone gets up from the Telly and plugs in the kettle to make tea. I assume that's what happened at the end of the wedding telecast.

A similar thing happened in the U.S. a couple of years ago. I think it was somewhere in New Mexico or Arizona that there was a pause in the super bowl game so a lot of people got up, went to the bathroom (all that beer) and flushed at nearly the same time which caused some sewer backups.

Don Chiasson

Security and dialbacks

<LIN@XX.LCS.MIT.EDU>
Fri, 25 Jul 1986 09:46 EDT

MSG: *MSG 5759
Date: 24 Jul 86 12:22:30 GMT
From: frog!die at EDDIE.MIT.EDU (Dave Emery, Software)
Re: Security and dialbacks
DISTRIB: *BBOARD

Summary: Dialbacks aren't very secure (repost of old article)
Apparently-To: codebreakers

In article <906@hoptoad.uucp> gnu@hoptoad.UUCP writes:
>Here are the two messages I have archived on the subject...

>[I believe the definitive article in that discussion was by Lauren Weinstein,
>vortex!lauren; perhaps he has a copy.

What follows is the original article that started the discussion. I do not know whether it qualifies as the "definitive article" as I think I remember Lauren and I both posted further comments.

- Dave

** ARTICLE FOLLOWS **

An increasingly popular technique for protecting dial-in ports from the ravages of hackers and other more sinister system penetrators is dial back operation wherein a legitimate user initiates a call to the system he desires to connect with, types in his user ID and perhaps a password, disconnects and waits for the system to call him back at a prearranged number. It is assumed that a penetrator will not be able to specify the dial back number (which is carefully protected), and so even if he is able to guess a user-name/password pair he cannot penetrate the system because he cannot do anything meaningful except type in a user-name and password when he is connected to the system. If he has a correct pair it is assumed the worst that could happen is a spurious call to some legitimate user which will do no harm and might even result in a security investigation.

Many installations depend on dial-back operation of modems for their principle protection against penetration via their dial up ports on the incorrect presumption that there is no way a penetrator could get connected to the modem on the call back call unless he was able to tap directly into the line being called back. Alas, this assumption is not always true - compromises in the design of modems and the telephone network unfortunately make it all too possible for a clever penetrator to get connected to the call back call and fool the modem into thinking that it had in fact dialed the legitimate user.

The problem areas are as follows:

Caller control central offices

Many older telephone central office switches implement caller control in which the release of the connection from a calling telephone to a called telephone is exclusively controlled by the originating telephone. This means that if the penetrator simply failed to hang up a call to a modem on such a central office after he typed the legitimate user's user-name and password, the modem would be unable to hang up the connection.

Almost all modems would simply go on-hook in this situation and not notice that the connection had not been broken. If the same line was used to dial out on as the call came in on, when the modem went to dial out to call the legitimate user back the it might not notice (there is no standard way of doing so electrically) that the penetrator was still connected on the line. This means that the modem might attempt to dial and then wait for an answerback tone from the far end modem. If the penetrator was kind enough to supply the answerback tone from his modem after he heard the system modem dial, he could make a connection and penetrate the system. Of course aome modems incorporate dial tone detectors and ringback detectors and in fact wait for dial tone before dialing, and ringback after dialing but fooling those with a recording of dial tone (or a dial tone generator chip) should pose little problem.

Trying to call out on a ringing line

Some modems are dumb enough to pick up a ringing line and attempt to make a call out on it. This fact could be used by a system penetrator to break dial back security even on joint control or called party control central offices. A penetrator would merely have to dial in on the dial-out line (which would work even if it was a separate line as long as the penetrator was able to obtain it's number), just as the modem was about to dial out. The same technique of waiting for dialing to complete and then supplying answerback tone could be used - and of course the same technique of supplying dial tone to a modem which waited for it would work here too.

Calling the dial-out line would work especially well in cases where the software controlling the modem either disabled auto-answer during the period between dial-in and dial-back (and thus allowed the line to ring with no action being taken) or allowed the modem to answer the line (auto-answer enabled) and paid no attention to whether the line was already connected when it tried to dial out on it.

The ring window

However, even carefully written software can be fooled by the ring window problem. Many central offices actually will connect an incoming call to a line if the line goes off hook just as the call comes in without first having put the 20 hz. ringing voltage on the line to make it ring. The ring voltage in many telephone central offices is supplied asynchronously every 6 seconds to every line on which there is an incoming call that has not been answered, so if an incoming call reaches a line just an instant after the end of the ring period and the line clairvointly responds by going off hook it may never see any ring voltage.

This means that a modem that picks up the line to dial out just as our penetrator dials in may not see any ring voltage and may therefore have no way of knowing that it is connected to an incoming call rather than the call originating circuitry of the switch. And even if the switch always rings before connecting an incoming call, most modems have a window just as they are going off hook to originate a call when they will ignore transients (such as ringing voltage) on the assumption that they originate from the going-off-hook process. [The author is aware that some central offices reverse battery (the polarity of the voltage on the line) in the answer condition to distinguish it from the originate condition, but as this is by no means universal few if any modems take advantage of the information supplied]

In Summary

It is thus impossible to say with any certainty that when a modem goes off hook and tries to dial out on a line which can accept incoming calls it really is connected to the switch and actually making an outgoing call. And because it is relatively easy for a system penetrator to fool the tone detecting circuitry in a modem into believing that it is seeing dial tone, ringback and so forth until he supplies answerback tone and connects and penetrates system security should not depend on this sort of dial-back.

Some Recommendations

Dial back using the same line used to dial in is not very secure and cannot be made completely secure with conventional modems. Use of dithered (random) time delays between dial in and dial back combined with allowing the modem to answer during the wait period (with provisions made for recognizing the fact that this wasn't the originated call - perhaps by checking to see if the modem is in originate or answer mode) will substantially reduce this window of vulnerability but nothing can completely eliminate it.

Obviously if one happens to be connected to an older caller control switch, using the same line for dial in and dial out isn't secure at all. It is easy to experimentally determine this, so it ought to be possible to avoid such situations.

Dial back using a separate line (or line and modem) for dialing out is much better, provided that either the dial out line is sterile (not readily traceable by a penetrator to the target system) or that it is a one way line that cannot accept incoming calls at all. Unfortunately the later technique is far superior to the former in most organizations as concealing the telephone number of dial out lines for long periods involves considerable risk. The author has not tried to order a dial out only telephone line, so he is unaware of what special charges might be made for this service or even if it is available.

A final word of warning

In years past it was possible to access telephone company test and verification trunks in some areas of the country by using mf tones from so called "blue boxes". These test trunks connect to special ports on telephone switches that allow a test connection to be made to a line that doesn't disconnect when the line hangs up. These test connections could be used to fool a dial out modem, even one on a dial out only line (since the telephone company needs a way to test it, they usually supply test connections to it even if the customer can't receive calls).

Access to verification and test ports and trunks has been tightened (they are a kind of dial-a-wiretap so it ought to be pretty difficult) but in any as in any system there is always the danger that someone, through stupidity or ignorance if not mendacity will allow a system penetrator access to one.

** Some more recent comments **

Since posting this I have had several people suggest use of PBX lines that can dial out but not be dialed into or outward WATS lines that also cannot be dialed. Several people have also suggested use of call forwarding to forward incoming calls on the dial out line to the security office. [This may not work too well in areas served by certain ESS's which ring the number from which calls are being forwarded once anyway in case someone forgot to cancel forwarding. Forwarding is also subject to being cancelled at random times by central office software reboots.]

And since posting this I actually tried making some measurements of how wide the incoming call window is for the modems we use for dial in at CRDS. It appears to be at least 2-3 seconds for US Robotics Courier 2400 baud modems. I found I could defeat same-line-for-dial-out dialback quite handily in a few dozen tries no matter what tricks I played with timing and watching modem status in the dial back login software. I eventually concluded that short of reprogramming the micro in the modem to be smarter about monitoring line state, there was little I could do at the login (getty) level to provide much security for same line dialback.

Since it usually took a few tries to break in, it is possible to provide some slight security improvement by sharply limiting the number of unsucessful callbacks per user per day so that a hacker with only a couple of passwords would have to try over a significant period of time.

Note that dialback on a dedicated dial-out only line is somewhat secure.

David I. Emery   Charles River Data Systems   617-626-1102
983 Concord St., Framingham, MA 01701.
uucp: decvax!frog!die

Search RISKS using swish-e

Report problems with the web pages to the maintainer

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 27

Tuesday, 29 July 1986

Contents

- Whoops! Lost an Area Code!
  Clayton Cramer
- Comet-Electra (RISKS-3.25)
  Stephen Little
- Comparing computer security with human security
  Bob Estell
- Info on RISKS (comp.risks)

Whoops! Lost an Area Code!

Clayton Cramer <voder!kontron!cramer@ucbvax.Berkeley.EDU>
Mon, 28 Jul 86 11:29:10 pdt

I had an interesting and aggravating experience this last Saturday. The 707 area code billing system failed. Completely. For over five hours.

During that time, you could not dial into the 707 area code, dial out of it, make local calls billed to a credit card, or get an operator. The ENTIRE area code. Fortunately, the 911 emergency number doesn't go through the billing system, so I doubt any lives were lost or threatened by this failure, but I shudder to think of how this could happen. My guess is someone cut over to a new release of software and it just failed.

No great philosophical comments, but one of those discouraging examples of the fragility of highly centralized systems.

Clayton E. Cramer

Comet-Electra (RISKS-3.25)

S Little <munnari!gucis.oz!edsel@seismo.CSS.GOV>

Tue, 29 Jul 86 15:14:30 est

Initial design studies for a trans-atlantic turbo-jet powered mail plane were begun during World War II by de Havilland. Eventually a much larger airliner, the DH-106 Comet prototype flew in 1949, so that computer involvement in the design is not an issue. The test program involved may have been adequate for forties technologies, but the jet-based mileages and altitudes obviously revealed a new range of problems which have resulted in the more stringent certification procedures now applied.

Whatever the source of the disastrous crack propagation (said in one case to be possibly a radio antenna fixing), the design change to rounded windows was in response to this danger. The only square window Comets remained in RAF service without pressurization for many years (Air International vol.12 no.4, 1977).

Given that computer representation is limited by our understanding of a design situation, is there a general concern with the performance of, inter alia, flight simulators, which may accurately represent an inadequate understanding of the behaviour of the system modelled. I have been told of one major accident in which the pilot followed the drill for a specific failure, as practiced on the simulator, only to crash because a critical common-mode feature of the system was neither understood, or incorporated in the simulation. I highly reccommend Charles Perrow's "Normal Accidents" for an analysis of the components of complexity in such situations.

I understand that the Shuttle auto-pilot is the source of re-appraisal including expert systems derivation of responses to the large number of relevant variables. What are people's feelings about the induction of knowledge in such areas, is it felt to increase or decrease risk via computer ?

Stephen Little, Computing & Information Studies,
Griffith Uni, Qld, Australia.

Comparing computer security with human security

"143B::ESTELL" <estell%143b.decnet@nwc-143b.ARPA>
29 Jul 86 08:29:00 PST

The question has been raised: Are there significant differences in the quality of security in computer system, based on elaborate software models [passwords, access lists, et al], versus having human guards at the door; e.g., humans can be bribed, computers can't; but computers can fail.

Hmmmmm... First let me admit a bias: I think the "MIT rule" applies:
No system can be better than the PEOPLE who design, build, and operate it.
[I call it that because that's where I first heard in in '68.]

Aside from that bias, there seems to be some assumptions:
(1) People don't "fail" [at least not like computers do]; and
(2) Computer can't be "diverted" in the manner of a bribe.

Seems to me that people DO FAIL, somewhat like computers; i.e., we have memory lapses [similar perhaps to incorrect data fetches?]; and we make perception errors [similar perhaps to routing replies to the wrong CRT?]

And computers can be diverted. Examples:

(1) A malicious agent, only wanting to deny others service on a computer, rather than gain access himself, can often find ways to exploit the priority structure of the system; e.g., some timesharing systems give high priority to "login" sequences; attacking these with a "faulty modem" can drain CPU resources phenominally.

(2) There are some operating systems/security packages that fail in a combination of circumstances; I'm going to be deliberatly vague here, in part because the details were shared with me with the understanding that I not broadcast them, and in part because I've forgotten them, and in part because the exact info is not key to the discussion; but to continue:

If the terminal input buffer is overrun [e.g., if the user-id or password is VERY long], and if the "next" dozen [or so] bytes matches a "key string" then the intruder is allowed on; not only that, but at a privileged level.

In other words, the code gets confused. But isn't that what a person suffers when he trades his freedom, his honor, and all his future earnings [hundreds of thousands of dollars?] for a few "easy" tens of thousands of dollars now for one false act? I'm saying that most "bribes" aren't nearly large enough to let the "criminal" relocate somewhere safe from extradition, and live a life of luxury ever after; instead, most bribes are only big enough to "buy a new car" or pay a overdue mortgage or medical bill.

OR is the real risk in both cases [human and computer] that the most potent penetrations are those that never come to light; e.g., the computer "bug" that is so subtle that it leaves no traces; and the "human bribe" that is so tempting that authorities [and victims] don't talk about it - precisely because they don't want folks to know how much it can be worth?

Discussion and comments, please. Bob

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 28

Thursday, 31 July 1986

Contents

- Laserprinter dangers
  Mansfiel
- Errors in error-handlers
  Mansfiel
- Military testing errors
  Alan Wexelblat
- Re: Comet-Electra (RISKS-3.25)
  Bill Fisher
- Computer and Human Security
  Lindsay Marshall
- Info on RISKS (comp.risks)

Laserprinter dangers

<MANSFIEL%DHDEMBL5.BITNET@WISCVM.ARPA>
Mon 31 Jul 86 17:38:10 N

Increasingly, large and "official" organisations such as motor vehicle tax offices, insurance companies, etc. are using laser printers to print the bills and other requests for money which are sent to customers. Whereas previously pre-printed letterheads (often with several and or coloured inks) were used, now the laser printer is relied on to print the letterhead itself, so that plain paper can be used.

It is probably only a matter of time before some clever person prints off a batch that looks fine but that have the c.d.'s own account number (or some other slightly safer one) on them, sends them out, and gets lots of money.

There must be lots of other forgery and swindling possibilities with laser printers. Have any frauds of this type have actually been committed?

[Most banks no longer make blank deposit slips routinely available, after various episodes of people magnetically coding account numbers onto the blanks and leaving these slips in the stack of blanks. Spoofing of

letterheads is of course relatively easy with laser printers, but also
with many of the electronic mailers around the net. PGN]

Errors in error-handlers

<MANSFIEL%DHDEMBL5.BITNET@WISCVM.ARPA>
Mon 31 Jul 86 15:47:17 N

Ken Laws, in RISKS-3.25 said

> Errors that arise within the error handlers are similarly
> important, but beyond my ability to even contemplate in
> the context of current languages.

A related problem, but much simpler and much more common in my experience, is that the user-written error handling code contains lots of errors. Reasons for this include

(a) This code is not considered "important", because we don't really expect it ever to be used, and even if it is, it will be used so rarely that normal criteria for neatness, etc., are not relevant.

(b) To exercise the code, the errors have to be caused or simulated. This is just too much work, especially as the program works "satisfactorily" as it is anyway.

The usual result is that when a rare error occurs, the error handler blows up, or worse, gives a wrong report. Then, having found the problem after many fevered days, you realise that the one time you need all the help you can get, including accurate error reports, is when you are under pressure to repair a crashed system, and you vow that in future ...

Military testing errors

Alan Wexelblat <wex@mcc.com>
Wed, 30 Jul 86 14:49:03 CDT

The following second-hand item appeared in the local Austin rag:

"SANTA ANA, Calif (AP) - A Pentagon error that knocked off two points on aptitude test taken by military recruits caused thousands of servicemen to lose training and benefits, according to a newspaper report.

The scoring error on nearly 2 million aptitude tests since 1984 could have been crucial for some recruits, because a single point can mean the difference between college-level training and a less-desirable assignment.

The _Orange County Register_ said Saturday that the military did not announce the errors but acknowledged them when queried by the newspaper. [...]

Rep. Robert Badham, R-Calif., said the House Armed Services Subcommittee on Military Personnel is investigating the testing problem and its effects.

It was unclear what caused the problem. The newspaper said that the error was apparently due to either to a miscalculation of the scoring curve incorporated into the Chicago testing computer or an actual misprint in the test booklets."

Does anyone have any better information than this?
Alan Wexelblat
ARPA: WEX@MCC.ARPA
UUCP: {ihnp4, seismo, harvard, gatech, pyramid}!ut-sally!im4u!milano!wex

"It is quite impossible for any design to be `the logical outcome of the requirements' simply because, the requirements being in conflict, their logical outcome is an impossibility."

Re: Comet-Electra (RISKS-3.25)

<bfisher.ES@Xerox.COM>
30 Jul 86 07:33:41 PDT (Wednesday)

Some years back (>10) there was a book out, "The Tail of the Comet," analyzing the design process for the Comet and then the investigations and procedures which pinpointed the design errors. I can't remember the author, but a comment of his is carved in memory, viz., "Extrapolation and interpolation are the fertile parents of error."

Bill Fisher

Computer and Human Security

"Lindsay F. Marshall" <lindsay%cheviot.newcastle.ac.uk@Cs.Ucl.AC.UK>
Wed, 30 Jul 86 13:30:00 gmt

I feel that there are significant differences between the quality of the two sorts of security. I appreciate the similarities that Bob has described and agree with his "MIT" rule, but there are many instances where computer security seems very much more superficial than human security. Passwords are the most obvious example - there is no simple way to determine whether or not the person typing the password is in fact the person expected, whereas there are other clues available to a trained human (NOT that I am saying that these are always correct or are always used!). In simplistic terms, it is much easier (for the average person) to impersonate someone "anonymously" by using their password, than it is for someone to actual pretend to be that person to other people. Of course, someone with enough confidence can get away with a phenomenal amount of pretence, because most people aren't really supicious (e.g., men in white coats in hospitals/labs, cleaners, postmen (cf. Father Brown story, "The Invisible Man")) or because people don't follow the rules (e.g. people with photos of apes/Einstein stuck to their identity cards). An example from my own experience when working in Industry:

I had received a tape written at 1600bpi on an IBM machine and needed a copy made at 800bpi for our PDP-11, so I went to the computer centre of our parent organisation, stopped an operator and asked him to make the copy and if possible to run the job that was on the tape. (It was an ENORMOUS Fortran H compilation...) The op said OK and I hung around a bit, looked over peoples shoulders and chatted with some people whom I knew, but that wasn't obvious). An hour later the op returned with my tapes and listing and said "By the way, who are you?". The day after that they installed electronic card locks on all the doors to the computing centre and stationed someone on the door....

I got away with this a) because I had never thought that there would be a problem, and so was the reverse of furtive (I may add that I had a lot of hair at time as well) and b) because the management hadn't actually considered the security risks (they did MOD work on the machine). On these lines has anybody more information about the Lockheed document scandal or is that too hush-hush???

Lindsay

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 29

Friday, 1 August 1986

Contents

- Ozone hole undetected for years due to programming error
  Bill McGarry
- Aircraft simulators and risks
  Art Evans
- Military testing errors
  Scott E. Preece
- Risks: computers in the electoral process
  Kurt Hyde via Pete Kaiser
- Risks of CAD
  Alan Wexelblat
- Info on RISKS (comp.risks)

Ozone hole undetected for years due to programming error

Bill McGarry <sdcsvax!dcdwest!ittatc!bunker!wtm@ucbvax.Berkeley.EDU>
Fri, 1 Aug 86 0:48:48 EDT

(I read the following in a magazine but when I went to write this article, I could not remember which magazine and some of the exact details. My apologies for any inaccuracies.)

Recently, it was disclosed that a large hole in the ozone layer appears once a year over the South Pole. The researchers had first detected this hole approximately 8 years ago by tests done at the South Pole itself.

Why did they wait 8 years to disclose this disturbing fact? Because the satellite that normally gives ozone levels had not reported any such hole and the researchers could not believe that the satellite's figures could be incorrect. It took 8 years of testing before they felt confident enough to dispute the satellite's figures.

And why did the satellite fail to report this hole? Because it had been programmed to reject values that fell outside the "normal" range!

I do not know which is more disturbing -- that the researchers had so much faith in the satellite that it took 8 years of testing before they would dispute the satellite or that the satellite would observe this huge drop in the ozone level year after year and just throw the results away?

Bill McGarry
Bunker Ramo, Trumbull, CT
{decvax, philabs, ittatc, fortune}!bunker!wtm

[A truly remarkable saga. I read it too, and was going to report on it -- but could not find the source. HELP, PLEASE! PGN]

Aircraft simulators and risks

"Art Evans" <Evans@TL-20B.ARPA>
Thu 31 Jul 86 13:26:48-EDT

In RISKS-3.27, Stephen Little comments on the risks in using an aircraft simulator which inadequately represents the aircraft being simulated:

> I have been told of one major accident in which the pilot followed the drill for a specific failure, as practiced on the simulator, only to crash because a critical common-mode feature of the system was neither understood, or incorporated in the simulation.

The implication is that use of such a simulator is risky, which is surely true. However, as is so often the case, we must also examine the risk of not using the simulator. Pilots flying simulators frequently practice maneuvers which are quite risky in a real aircraft. A common example is loss of power in one engine at a critical moment on takeoff. This is just too risky to practice for real (since sometimes the "right" answer is to crash straight ahead on the softest and least expensive piece of real estate in sight), but practice in the simulator is quite valuable. All we can do is make the simulator as good as state of the art permits, and improve it whenever we are subjected to one of the expensive lessons Little refers to.

Little also comments on the shuttle simulator. There, I would guess, the critical issue is the cost of using the real thing as opposed to cost of the simulator. Again, the simulator is as good as practical, and is improved as more data are gathered.

Art Evans

Military testing errors

"Scott E. Preece" <preece%ccvaxa@GSWD-VMS.ARPA>
Fri, 1 Aug 86 16:53:13 cdt

The New York Times report indicated that some of the tests were printed with a major section set in six-point type instead of ten-point, making it very hard to read. The section consisted of math word problems and the object was to do as many as possible in a set time. People with the small-type tests did significantly worse than those with the large-type tests. Although this MIGHT be a computer-related problem (if the error was, for instance, lack of a font change in a machine-readable source file), I don't think the article specifically said that.

Risks: computers in the electoral process

Systems Consultant <kaiser%furilo.DEC@decwrl.DEC.COM>
01-Aug-1986 1529

There will be a symposium on security and reliability of computers in the electoral process at Boston University this August 14th & 15.

Computers are relatively new in the electoral process and most decision makers in this process have little, if any, experience. One of the speakers found evidence of a Trojan Horse in ballot counting software. He will be speaking about that in the symposium.

PLACE: Boston University Engineering Building, Room B33
DATE: August 14th & 15th
TIME: 9:00 AM thru 4:00 PM

I would like to thank the many RISKS readers who contributed last semester to my students' request for ideas on how to make the computerized voting booth safe from computer fraud. I'll be presenting many of the findings of our study.

Kurt Hyde

[Recall Ron Newman's detailed summary in RISKS-2.42 of Eva Waskell's talk on this subject. Perhaps we will get an update on any new information presented at BU. We look forward to Kurt's findings as well. PGN]

Risks of CAD

Alan Wexelblat <wex@mcc.com>
Fri, 1 Aug 86 15:45:54 CDT

Henry Petroski's book, _To Engineer is Human_ contains a chapter called "From Slide Rule to Computer," in which he talks about some risks of computers and specifically of computer-aided design (CAD). I will try to summarize his main points below.

Petroski points out that the transition away from slide rules has, in itself, some risks. First of all, there is the problem of precision. Everyone knows that computers can produce very precise results, but this tends to blind us to the fact that the results are really no more precise than the inputs that were combined to produce them. A twelve-digit answer is no good if one of your inputs is accurate to only three digits.

A side effect of this is that we have tended to lose a `feel' for the proper magnitudes for our numbers. When arithmetic was done on a slide rule, students had to supply the decimal place and thus needed to know approximately how big the answer should be. This lack of feel seems to have been (at least part of) the problem with the x-ray machine that burned a patient by applying too large a dose.

In "the old days" calculating stresses and the like was expensive and so engineers didn't have time to do too much of it. So they tended to design things that were close to their experience and where they knew approximately what the stresses, etc. should be. With optimization (and other CAD) packages, engineers can do much more calculating and can therefore design structures that are more novel and that they are less familiar with. This increases the risk that the engineer will not be able to spot errors in the CAD programs' output. Again, he has no `feel' for what the output should be.

Petroski also fears that inadequate computer simulation is replacing crucial real testing. Engineers who are not programmers may not realize that certain stress calculations have not been done by the program; thus he may be inclined to forgo simple things (like physically stretching or bending a pipe to see where it breaks). An example of this oversimplification is the collapse of the roof of the Hartford Civic Center (under a weight of ice and snow). Post mortem analysis revealed that the interconnection of the rods and girders in the ceiling had been modeled too simplistically in the computer programs that were used during the design.

In general, Petroski fears that the CAD programs' optimization of things is leading to structures that are "least-safe." That is, there's no room for error in the optimized structure.

There is also a risk that with a software crutch a less-than-qualified engineer can put together a design that looks better than it is. Even an engineer who is qualified in one area may be encouraged by the ease of CAD to venture outside his area of expertise.

There is also one other item of interest to RISKS readers. In the chapter called "The Limits of Design," Petroski quotes from the proceedings of the "Proceedings of the First International Conference on Computing in Civil Engineering." Apparently, there was a session on `Computer Disasters' at that conference, but NO PAPERS WERE PUBLISHED. Supposedly, this encouraged candor. The conference was held in New York, in 1981. Were any RISKS readers there? Do you know someone who was? It would be interesting to see if we can construct a list of our own.

In any event, Petroski's book (ISBN 0-312-80680-9) is a good read and can be bought at a discount by members of LCIS. I recommend it highly.

Alan Wexelblat
UUCP: {ihnp4, seismo, harvard, gatech, pyramid}!ut-sally!im4u!milano!wex

Search RISKS using swish-e

Report problems with the web pages to the maintainer

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 30

Monday, 4 August 1986

Contents

- Ozone hole undetected
  Jeffrey Mogul
- Re: Risks of CAD
  Henry Spencer
- Comment on Hartford Civic Roof Design
  Richard S D'Ippolito
- Expert system to catch spies
  Larry Van Sickle
- Info on RISKS (comp.risks)

Ozone hole undetected

Jeffrey Mogul <mogul@decwrl.DEC.COM>
4 Aug 1986 1058-PDT (Monday)

Although I, too, am relying on memory, I'm pretty sure that the article Bill McGarry mentioned was published in The New Yorker sometime during the past two or three months. [Also something in Science a few issues ago on the phenonenon itself... PGN]

My understanding is that it was not so much a case of the researchers believing the satellite instead of other evidence, but rather that the researchers who ran the satellite must not have been too terribly interested in what was going on over the poles. After all, if they were interested, I would think they might have been bothered by large empty spots in their data.

As to Bill's being disturbed that "the satellite would observe this huge drop in the ozone level year after year and just throw the results away", I think this imputes a certain level of intelligence to the computer system that probably isn't there. I'd bet that their computer spits out maps of the ozone layer, but probably doesn't have any facility to spot trends.

Still, it's obvious that a little more care in the decision to discard anomalous data would have gone a long way. When humans through away anomalous results, at least they realize that they are doing so [although not always consciously; see Stephen Jay Gould's "The Mismeasure of Man".] When a computer throws away anomalous data, the user might not be aware that anything unusual is going on. A good program would at least remark that it has thrown away some fraction of the input data, to alert the user that something might be amiss.

Re: Risks of CAD

<decwrl!decvax!utzoo!henry@ucbvax.Berkeley.EDU>
Sun, 3 Aug 86 03:17:32 edt

Alan Wexelblat comments:

> Petroski also fears that inadequate computer simulation is replacing crucial
> real testing...

One can see examples of the sort of engineering this produces in many pieces of high-tech US military equipment. In the recent times, the criteria used to evaluate a new military system have increasingly drifted away from straight field-test results and toward complex and arbitrary scoring schemes only vaguely related to real use. Consider how many official reports on the Sergeant York air-defence gun concluded, essentially, "no serious problems", when people participating in actual trials clearly knew better. Some of this was probably deliberate obfuscation -- juggling the scoring scheme to make the results look good -- but this was possible only because the evaluation process was well divorced from the field trials. Another infamous example is the study a decade or so ago which seriously contended that the F-15 would have a kill ratio of several hundred to one against typical opposition. These are conspicuous cases because the evaluation results are so grossly unrealistic, but a lot of this goes on, and the result is unreliable equipment with poor performance.

It should be noted, however, that there is "real testing" and real testing. Even the most realistic testing is usually no better than a fair facsimile of worst-case real conditions. The shuttle boosters superficially looked all right because conditions had never been bad enough to produce major failure. The Copperhead laser-guided antitank shell looks good until you note that most testing has been in places like Arizona, not in the cloud and drizzle more typical of a land war in Europe. Trustworthy test results come from real efforts to produce realistic conditions and vary them as much as possible; witness the lengthy and elaborate tests a new aircraft gets. Even if the results of CAD do get real-world testing, one has to wonder whether those tests will be scattered data points to "validate" the output of simulations, as opposed to thorough efforts to uncover subtle flaws that may be hiding between the data points.

Henry Spencer @ U of Toronto Zoology
{allegra,ihnp4,decvax,pyramid}!utzoo!henry

Comment on Hartford Civic Roof Design

<Richard.S.D'Ippolito@sei.cmu.edu>
4 Aug 1986 00:33:41-EDT

I would like to point out that Alan Wexelblat's comment on inadequate use of computers for CAD might be somewhat misleading regarding the roof modelling for the Hartford Civic Center. The problem was that the program user selected the wrong model for the beam connection to be used. When the program was re-run with the correct model, it predicted the collapse in precisely the mode that it happened. I'm not sure that that was clear from the wording in Mr. Wexelblat's comment, i.e., that the modelling was improperly done by the operator (GIGO again!).

Richard D'Ippolito, P.E.
Carnegie-Mellon University
Software Engineering Institute
(412)268-6752
rsd@SEI.CMU.EDU

<CS.VANSICKLE at R20.UTEXAS.EDU>
Wednesday, 23 July 1986 22:39-EDT

Today's (July 23, 1986) Wall Street Journal contains an editorial by Paul M. Rosa urging the use of expert systems to identify potential spies (acutally traitors). Mr. Rosa is a lawyer and a former intelligence analyst. Since virtually all American traitors sell out for money, an expert system embodying the expertise of trained investigators could examine credit histories, court files, registers of titled assets such as real estate and vehicles, airline reservations, telephone records, income tax returns, bank transactions, use of passports, and issuance of visas. The system would look for suspicious patterns and alert counter-intelligence officials for further investigation.

There are some obvious considerations of privacy and legality, but that is probably best discussed on another bulletin board. Mr. Rosa says the system would be used only on the 4.3 million people who hold security clearances, who have consented to government scrutiny.

According to Mr. Rosa, "the obstacles to implementation are not technological," and "the system could be implemented quickly and cheaply." He predicts that the Soviets, working through their extensive international banking network, will use the same techniques to identify potential recruits. He also says that the FBI has three expert systems for monitoring labor rackets, narcotics shipments, and terrorist activities.

Any reactions? Is this doable? It strikes me as more of a data collection problem than an expert system problem. Is there anyone who knows more about the FBI expert systems and can talk about it?

Larry Van Sickle

cs.vansickle@r20.utexas.edu
Computer Sciences Dept.
U of Texas at Austin
Austin, TX 78712

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 31

Tuesday, 5 August 1986

Contents

- Another cruise missile lands outside Eglin test range
  Martin J. Moore
- Aircraft simulators and risks
  Gary Wemmerus
- Re: Comment on Hartford Civic Roof Design
  Brad Davis
- Expert system to catch spies (RISKS-3.30)
  Chris McDonald
- Computer and Human Security
  Henry Spencer
- Ozone Reference
  Eugene Miya
- Financial risks
  Robert Stroud
- Mail Load Light(e)ning?
  SRI-CSL Mail Daemon
- Info on RISKS (comp.risks)

Another cruise missile lands outside Eglin test range

<mooremj@eglin-vax>
0 0 00:00:00 CDT

An unarmed Tomahawk cruise missile malfunctioned and landed unexpectedly during a test launch at Eglin AFB last Saturday (8/2/86). The missile, launched from the battleship Iowa at 10:15 am CDT, flew successfully for 69 minutes before deploying its recovery parachute for reasons not yet determined. The missile made a soft landing in an uninhabited area 16 miles west of Monroeville, Alabama. No injuries or property damage were reported.

The cause of the failure is not yet known. The missile, which suffered no apparent external damage, was recovered and returned to the General Dynamics works in San Diego for investigation. The missile was the second in four launches to land outside the 800-square-mile Eglin reservation. Last December

8, the first Tomahawk launched at Eglin landed near Freeport, Florida. The cause of that failure was a procedural problem which caused portions of the missile's flight control program to be erased during loading.

Saturday's failure followed a successful Tomahawk launch on the previous day. A missile launched from the destroyer Conolly successfully flew a 500-mile zigzag course over southern Alabama and the Florida Panhandle before landing at the designated recovery point on the Eglin range.

-- Martin J. Moore

To: Art Evans <Evans@tl-20b.arpa>
cc: Risks@csl.sri.com
Subject: Re: Aircraft simulators and risks
Date: Tue, 05 Aug 86 09:45:51 -0800
From: Gary Wemmerus <gfw@ICSE.UCI.EDU>

I heard a story about the DC-10 crash at O'Hare in 1979 that might be the one you mentioned.

After the crash, they programmed that sequence of events into the simulator and tried out pilots on it. Every one of the pilots that followed the correct procedures as listed in the MANUAL for that sequence of events CRASHED. The problem was that the sequence of events did not include loss of an engine, just loss of engine power, and did not take into account total loss of hydraulic power. I have heard that there are no instruments on the DC-10 that would tell a pilot that the engine was gone, just that there was no power from it.

When pilots tried a different way or responding to the sequence of events, I believe that a successful landing was achieved 80% of the time. I think that there was no problem with the simulator, but there were two sets of events that led to one set of indicators to the pilot, and the manual listed the correct procedure for the other set of events. My guess is that they never expected the sequence that occurred and have now come up with a way to distinguish between the two events.

-gfw

PS. A lot of this is from second-hand sources, so I cannot totally vouch for its accuracy.

Re: Comment on Hartford Civic Roof Design

Brad Davis <b-davis@utah-cs.arpa>
Tue, 5 Aug 86 13:18:08 MDT

Along with the problems of wrong model is the problems with not testing at proper extremes or making bad assumptions. About 15 years ago a new shopping mall was being built in Salt Lake City. The engineers (and architects?) from California consulted their data books (or ran their CAD systems) and determined the amount of weight

the building needed to support to make it through a desert winter.
Even though Utah is a desert, we get 1 foot snowfalls in twelve-hour periods. The roof caved in at the first big snowfall of the season. Luckily the mall hadn't opened yet. They did fix it and the mall hasn't had any problems since.

Brad Davis

Expert system to catch spies (RISKS-3.30)

Chris McDonald SD <cmcdonal@wsmr06.arpa>
Tue, 5 Aug 86 7:31:33 MDT

Larry Van Sickle asks the question "Is it doable?" regarding the use of an "expert system" to screen out or to identify potential espionage agents. From my sixteen years of experience in positions which require a security clearance and actually access to classified defense information, I conclude "NO!" The reason is that potentially millions of government as well as contractor employees have clearances with access to national defense information. I find it incredible to belive that any "expert system" could realistically factor in all the variables which might cause an individual to be recruited for espionage or to recruit him or herself for such activity.

Second, while the news media has reported the apparent "greed" of the most recent batch of US citizens involved in espionage against their country, I would surmise that there were probably equally compelling personnel and philosophical reasons for their actions. Whenever there is an in-depth damage assessment of espionage cases "after the fact," it seems historically that there are many motivations at work.

Third, if "disaffection" might be one of the causes of a successful espionage recruitment, then the problem is magnified by the very bureaucracy that employs individuals with security clearances. For example, there has not been a President or Executive Branch since 1970 which has not proposed that the Federal workforce is a collection of lazy, misfits who could not be employed anywhere else. There has never been a sustained call for "excellence" in the government on the assumption that this is a contradiction in terms. How could any "expert system" factor in cuts in salary, retirement and benefits without-- perhaps with some exaggeration-- potentially disqualifying the entire workforce. The defense contractor side of the house experiences the same sort of problems as it goes through one cycle after another in which today we build the B-1 bomber and the next day we shut down the line.

Finally, although I do not have the benefit of reading the actually article which Larry mentions, it does appear that the so-called "former intelligence analyst" has confused the issues of "suitability" and "loyalty". Just because an individual has financial problems does not necessarily mean that he will spy against the US. While "suitability" factors may appear in actual espionage cases to have had some influence on "loyalty," they are usually never the sole reason. Indeed, if "greed" alone were a factor, why have so many people "sold" themselves so cheaply?

Date: Tue, 5 Aug 86 21:41:12 edt
From: decwrl!decvax!LOCAL!utzoo!henry@ucbvax.Berkeley.EDU
To: LOCAL!CSL.SRI.COM!RISKS
Subject: Computer and Human Security

Lindsay F. Marshall writes, in part:

> I feel that there are significant differences between the quality of the two
> sorts of security... there are many instances where computer
> security seems very much more superficial than human security...

The other side of this coin is that there are many instances where human security is very much more superficial than computer security. How many times have you been waved through a gate by a guard who knows you? Does he really consider the possibility that your pass might have been revoked yesterday? Yes, I know, they're supposed to always check, but it often doesn't work that way in practice. Especially if there is something else distracting them at the time. An electronic pass-checker box, on the other hand, does not get distracted and doesn't get to know you. Human security can be bribed, coerced, or tricked; these tactics generally don't work on computers. Their single-minded dedication to doing their job precisely correctly and ignoring everything else blinds them to "out-of-band" signs that subversion is taking place, but it also blinds them to "out-of-band" methods of subversion.

The best approach is to combine the virtues of the two systems: use computers for mindless zero-defects jobs like checking credentials, and use humans to watch for improper use of credentials, attempts to bypass credential checking, and anomalies in general. One gray area is checking the match between credentials and credential-holders: this generally has to be done by humans unless the credentials are something like retinagrams.

Henry Spencer @ U of Toronto Zoology
{allegra,ihnp4,decvax,pyramid}!utzoo!henry

Ozone Reference

Eugene Miya <eugene@AMES-NAS.ARPA>
Tue, 5 Aug 86 10:51:50 pdt

I talked to one of our bio-geo-chemists. There is a popular article which he feels is a good introduction to the players of this research including good references:

Nature, 321, June 19, 1986, pp. 729-730

To reiterate: all of the postings I have seen on Risks almost make this sound like either a conspiracy or foot dragging by the earth science community. Eight years is nothing in the span of research in the earth sciences. That was also the length of time involved in the Palmdale Bulge research which turned out to be erroneous.

My contact, Greg, has seen papers suggesting natural mechanisms for ozone depletion in the Antarctic. There is insufficient money and time to research long-period phenomena. Note: this brings up the issue of fast developing trends with slow thinking scientific communities, but that is another issue.

--eugene miya, NASA Ames

[The AAAS Science article is on page 1602 of the 27 June 1986 issue. It points out the increasing depletion (now 50%) in the ozone layer for a short period in October compared with the 1979 norm. It does not deal with the reported software problem. PGN]

Financial risks

Robert Stroud <robert%kelpie.newcastle.ac.uk@Cs.Ucl.AC.UK>
Tue, 5 Aug 86 16:17:45 bst

There was an item on the ITV News at Ten last night about the record 62-point fall of the Dow Jones Index about a month ago. Since it was on TV, I can't report it verbatim, but the gist was as follows:

"Experts are convinced that the record fall was almost entirely due to the use of computer programs that automatically sell stock when certain conditions are triggered. [...stuff about the cash index falling below the futures index...] Whereas a fall of this magnitude would have been disastrous a few years ago, nowadays it hardly causes a hiccup. The big shareholders are quite capable of withstanding a swing of 40 points or more in a day, although the small investor suffers. Although computers are blamed for this sort of instability, they are also credited with keeping the market at its high level over the last 6 months. However, members of the public would be concerned if they were aware of the increasing use of technology, not just because of the problems of the small investor but also because decisions are now being taken based solely on movements within the market, without consideration of external economic factors."

I also saw something in The Times suggesting that the fall was "aggravated" by the use of such programs a few days after the incident occurred - maybe ITV were reporting the result of an investigation into the causes.

There has been a recent trend towards relaxing controls and regulations in the financial markets. There will shortly be what is known as the Big Bang in the UK and this has caused a great deal of activity in the City with companies that have traditionally performed separate functions being allowed to merge, and several giant financial organisations forming. There has been a lot of headhunting with astronomical (by British standards :-) salaries being offered, first for dealers but more recently for those with computing experience. Sophisticated computer systems are planned, and apart from just displaying information, I expect there will be more programs to buy and sell automatically. Another aspect of the mergers will be the need to establish what are called Chinese Walls within institutions to prevent the unethical use of confidential information. For example, one part of an institution may be giving financial advice to some company which another part of the same institution could use to speculate - the same institution would not have been allowed to perform both roles under the regulations before the Big Bang.

The Chinese Wall problem is really a standard security problem with the computing system being divided up into multiple partitions between which information flow is not allowed. Human leakage is likely to be more of a problem. Increasing dependence on technology has obvious reliability implications, but I am more concerned about whether automatic trading is likely to have a destabilising influence. Modern telecommunication has made it possible to have a 24 hour world currency futures market in which vast sums (1 billion/day) are traded rapidly for minute gains. This is pure speculation, creating money out of nothing with no connection to the outside world, (unlike other futures markets which at least have some basis in reality providing a guaranteed market for some commodity). I feel that programs will be able to react too quickly for the wrong reasons with possibly disastrous consequences. Equally, they could create a false sense of security and an artificially inflated market by buying instead of selling.

Although some of these concerns are political rather than technical, and I am in no sense a financial expert, I would appreciate a discussion of these issues and some information about the heuristics and safeguards built into these automatic trading programs.

Robert Stroud, Computing Laboratory, University of Newcastle upon Tyne.

ARPA robert%cheviot.newcastle@ucl-cs.ARPA
UUCP ...!ukc!cheviot!robert
JANET robert@newcastle.cheviot

The Mailer Daemon <Mailer@CSL.SRI.COM>
Tue 5 Aug 86 19:37:04-PDT

Message undelivered after 14 days -- will try for another 1 day:
RISKS@DOCKMASTER.ARPA: Cannot connect to host

[The Dockmaster IMP was hit by lightning several weeks ago. It still has not recovered. The thundering of undelivered mail messages rains down upon me as my mailer merrily retries at intervals. PGN]

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 32

Wednesday, 6 August 1986

Contents

- DC-10 Crash
  Chuck Weinstock
- Earthquake Reporting
  AP
- The Recent Near-Disaster for the Shuttle Columbia
  Peter G. Neumann
- Traffic lights in Austin
  Alan Wexelblat
- Re: Laserprinter dangers
  Graeme Hirst
- Info on RISKS (comp.risks)

DC-10 Crash

<Chuck.Weinstock@sei.cmu.edu>
6 Aug 1986 09:19-EDT

I have also heard the stories about pilots following the procedures in the manual not being able to save the aircraft. In the case of the American Airlines DC-10 accident, the pilot executed the correct maneuver for loss of engine power, but the effects of the missing engine caused it to go into a stall. However, the correction for the stall is 180 degrees different from the correction for the loss of engine power, and thus the plane was lost. The pilot possibly could have saved the aircraft had he known what was going on. The reason the pilot didn't correct for the stall is that he didn't know about it (or knew too late) -- because the missing engine supplied power to the stall warning device.

Interestingly, at the time stories were circulating that some airlines (e.g., United) had ordered their DC-10's with dual-redundant stall-warning devices, powered off of multiple engines.

(I'm afraid I don't have a reference. Probably Aviation Week and Space Technology.)

Chuck

Earthquake Reporting

Peter G. Neumann <Neumann@CSL.SRI.COM>
Wed 6 Aug 86 11:55:55-PDT

From the AP, Tuesday, 5 August 1986, Los Angeles:

Three of five earthquakes that state agencies said rattled California on Sunday never happened, officials acknowledged yesterday. The false reports by California's Office of Emergency Services and Department of Water Resources were blamed on static in the microwave system that transmits data from monitoring devices around the state to Sacramento. Don Irwin, deputy director of Emergency Services, said his agency was trying to decide whether to change procedures and stop publicizing what he termed ``preliminary, unofficial information''.

U.S. Geological Survey seismologists said yesterday that three small quakes shook the state on Sunday, two near San Jose and a third in the eastern Sierra Nevada. No damage or injuries were reported. The state agencies never reported one of the San Jose-area quakes, and reported three others that did not happen.

The Recent Near-Disaster for the Shuttle Columbia

Peter G. Neumann <Neumann@CSL.SRI.COM>
Wed 6 Aug 86 13:22:33-PDT

From the San Francisco Chronicle, Wednesday, 6 August 1986:

WASHINGTON - The space shuttle Columbia (the launch preceding the Challenger disaster) came within 31 seconds of being launched without enough fuel to reach its planned orbit on January 6 after weary Kennedy Space Center workers mistakenly drained 18,000 gallons of liquid oxygen from the craft, according to documents released yesterday by the White House panel that probed the shuttle program. Although [NASA] said at the time that computer problems were responsible for the scrubbed launch, Representative Bill Nelson, D-Fla., who flew on the mission, said yesterday that he was informed of the fuel loss while aboard the spacecraft that day...

According to the appendix [to the panel report], Columbia's brush with disaster ... occurred when Lockheed Space Operations Co. workers "inadvertently" drained super-cold oxygen from the shuttle's external tank 5 minutes before the scheduled launch. The workers misread computer evidence of a failed valve and allowed a fuel line to remain open. The leak was detected when the cold oxygen caused a temperature gauge to drop below approved levels, but not until 31 seconds before the launch was the liftoff scrubbed.

NASA said then that the liftoff was scrubbed [until January 12] because computer problems delayed the closing of a valve. Space agency spokeswoman Shirley Green said yesterday that the fuel loss did not become apparent until much later.

The NY Times (same day) noted that the potentially catastrophic launch of the Columbia without adequate fuel to reach its intended orbit could be blamed on human error caused by fatigue. "Investigators also concluded that many key people working for NASA and its contractors work an excessive amount of overtime that has the potential for causing catastrophic errors in judgment."

The Chronicle article goes on to state, quoting the panel report, that fatigue may also have contributed "significantly" to the disputed decision by NASA and Thiokol officials to launch the Challenger in cold weather -- despite strong evidence that the O-ring booster seals were ineffective. The panel said "certain key managers obtained only minimal sleep the night before the teleconference" in which the fatal decision was made. Furthermore, a study of 2900 workers' timecards in the weeks before that showed an "unusally high amount of overtime", during which time there were five aborted launches and two actual launches.

I am astounded to look back over my list of computer-related disasters (an update will appear in RISKS at the beginning of Volume 4 -- it is now up to 5 pages) and find only one other space/missile/defense/aviation case that could easily have been linked to fatigue. That case was the KAL 007, whose real cause is still a matter of much speculation. (See ACM Software Engineering Notes 9 1 and 10 3.) One would expect that to be a more common cause...

Traffic lights in Austin

Alan Wexelblat <wex@mcc.com>
Wed, 6 Aug 86 14:01:12 CDT

Yesterday, Austin experienced a sudden thunderstorm and some small power failures. One of the things knocked out by the power loss was the central computer that coordinates the traffic lights in the downtown area.

The central controller is backed up by isolated controllers at each intersection. By my guesstimate, there are about 125 of these intersections. Two of the site controllers failed to operate, causing the light at those two intersections to go out.

Is this a success or a failure for the system as a whole? Of course we'd like it if the backup was 100%, but is 2% an acceptable failure rate?

(Side note: the only adverse effect of the two failures was that humans -- policemen - were required to stand in the downpour and direct traffic.)

Alan Wexelblat
UUCP: {ihnp4, seismo, harvard, gatech, pyramid}!ut-sally!im4u!milano!wex

[Success -- like failure -- is relative. Even the greatest successes can be disasters if we become overconfident. Even the worst disasters can have some benefits if we learn from them. In this case, the result was clearly a qualified success, but would have been quite different if someone had been killed when the lights went out at one intersection. PGN]

Re: Laserprinter dangers

Graeme Hirst <gh%ai.toronto.edu@CSNET-RELAY.ARPA>
Tue, 5 Aug 86 15:27:52 edt

> Increasingly, large and "official" organisations [...] are using laser
> printers to print the bills and other requests for money [...]

I cannot believe this will be a serious problem. In fact, most organizations are still using pre-printed stock, even if they use the laser printer to do smarter things on it. For example, my Ontario motor vehicle registration is laser-printed on banknote-style paper. My credit card bills and bank statements are laser-printed on pre-printed paper that is virtually identical in design to the paper used when they were impact-printed. (This also has programming advantages.)

Similarly, a new ATM at my bank prints its receipts on a role of paper like that of a cash register, instead of the pre-printed cards used by older models. But the paper used has the bank's logo printed on the back to prevent easy forgery.

The one exception I can think of is my city tax and water bills, which have (on plain colored paper) the most ornate laser-printing imaginable -- which required some amazing hacking on the Xerox 9700. Duplicating this would be of the same level of complexity as forging pre-printed stock -- which was always possible even in the days of hand-writing and typewriters.

\\\\ Graeme Hirst University of Toronto Computer Science Department
//// utcsri!utai!gh / gh@ai.toronto.edu / 416-978-8747

Search RISKS using swish-e

Report problems with the web pages to the maintainer

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 33

Thursday, 7 August 1986

Contents

- Air traffic computer failure
  Hal Perkins
- Re: Laserprinter dangers
  Sean Malloy
- Re: Expert system to catch spies
  Rich Kulawiec
- Survey of Computer Professionals
  Kurt Hyde
- Info on RISKS (comp.risks)

Air traffic computer failure

Hal Perkins <hal@gvax.cs.cornell.edu>
Fri, 8 Aug 86 00:14:04 EDT

From the New York Times, Thursday, August 7, 1986, p. A10.

Computer Failure Snarls Chicago Air Traffic

WASHINGTON, Aug. 6 (UPI) -- The main computer used by air traffic controllers at Chicago Center, the Federal Aviation Administration's busiest facility, failed Tuesday, delaying hundreds of flights, an agency spokesman said today.

The failure, which lasted two hours, during which a backup computer operated, caused no safety-related incidents, a spokesman, Robert Buckhorn, said.

The incident at 2 P.M. was caused by the failure of a computer element that feeds the computer radar information and other data critical to tracking and directing flights in the crowded Chicago airspace, agency sources familiar with the breakdown said.

In Chicago, agency sources said some of the main computer's functions were restored Tuesday afternoon. Mr. Buckhorn said the problem was completely corrected at about 6 A.M. today.

[Anybody know further details about this? HP]

Re: Laserprinter dangers

Sean Malloy <malloy@nprdc.arpa>
Thu, 7 Aug 86 06:55:10 pdt

>From: Graeme Hirst <gh%ai.toronto.edu@CSNET-RELAY.ARPA>
>Subject: Re: Laserprinter dangers
>
>The one exception I can think of is my city tax and water bills, which have
>(on plain colored paper) the most ornate laser-printing imaginable -- which
>required some amazing hacking on the Xerox 9700. Duplicating this would be
>of the same level of complexity as forging pre-printed stock ...

This is less of a problem than you might imagine -- Any good laser printer has a page control language, such as PostScript on the Imagen laser printer at my office, that can output bitmap images. And with the availability of graphic input devices like digitizing cameras and image scanners, the problem of entering ornate output formats is due more to the price of the input devices than the actual input itself.

And even if you have to put the paper through twice, once for the fixed ornate work, and once for the text of the bill itself, the result is going to look like the real thing. And with some of the page layout packages like InterLeaf, the whole output can be laid up for each page on a single pass, at the expense of speed of output (InterLeaf eats an amazing amount of CPU time).

Simply having a complex output format isn't enough to prevent forgery -- all that will happen is that the forgers will have to resort to the same technology that created the image in the first place.

Sean Malloy, Naval Personnel R&D Center, malloy@nprdc

Re: Expert system to catch spies

Whitewater Wombat <rsk@purdue-asc.ARPA>
Thu, 7 Aug 86 22:57:24 est

Mr. Rosa's recommendation that expert systems be used in order to identify potential spies certainly has some chilling Orwellian overtones, and also highlights certain misconceptions about expert systems.

The cross-correlation of credit histories, bank records, major purchase receipts, customs logs, and so on, is certainly a monumental task, given the size of the databases involved if such a program were applied on a national scale; but this sort of problem seems to me to be within the reach of ordinary database query systems. In my opinion, a program which performs such searching operations is not an expert system, but a (smart) database manager. Calling it an expert system does not make it one.

Chris McDonald points out another important problem; "suitability", in terms of whatever criteria are employed, does not necessarily imply guilt. For example, if I were to design the criteria, I might direct the program to search for frequent overseas travellers with multiple bank accounts and expensive automobiles. Of course, the resultant list of "suspects" would be huge, and would probably contain a great number of prominent business executives. Certainly, this is a facetious example, but extending and refining the criteria will only partially reduce the list. Given the initial (huge) size of the search space, I wonder whether the reductions would ever be sufficient to reduce it to a humanly-manageable size. I speculate that a case-by-case examination of the list would simply not be feasible.

Finally, the public at large (apparently including Mr. Rosa) does not seem to understand that expert systems are built to embody the knowledge of human experts. (Perhaps this will eventually change; but I am as yet unaware of any self-taught expert system.) System architects spend a great deal of time querying human experts to find out how they reason about the problem space, and then attempt to construct a system that (loosely) mimics that process. To a large extent, the efficacy of an expert system depends upon the expertise of those whose collective experiences were tapped to build it. If a spy-catching expert system is to be reasonably successful, then at least one human expert must be found...but is there one? Is there at least one person whose acumen is comparable with, say, the medical diagnostic skills of the physicians involved in the Mycin project?

My intuition says that there is not. (But I'll hedge my bets by observing that if the U.S. government actually had such a person in their employ, they'd be unlikely to publicize that fact.) It seems to me that Mr. Rosa is invoking the modern magic buzzword "expert system" as if he expects a team of software engineers to solve national security problems for him. Given the limited (impressive, but limited) success that expert systems have enjoyed in such highly restricted problem domains as mineralogical prospecting and computer system configuration, I doubt that they'd be much help in such a wide-open area as espionage.

Rich Kulawiec, pucc-j!rsk, rsk@j.cc.purdue.edu, rsk@purdue-asc.arpa

---

Survey of Computer Professionals [REPLY TO KURT, NOT RISKS]

Kurt Hyde DTN 264-7759 MKO1-2/E02 <hyde%vax4.DEC@decwrl.DEC.COM>
Thursday, 7 Aug 1986 07:32:41-PDT

Survey of Computer Professionals Regarding Computerized Voting

Please return to TOPCAT::HYDE on Digital's Engineering Net by

Tuseday, August 12th.

1) Would you trust a computerized voting system if did not allow you
   to monitor how it worked nor did it allow you to inspect the ballot
   it cast for you?

   YES, I would trust it   NO, I not would trust it

2) Would you trust a computerized voting system if did allow you to
   monitor how it worked, but did not allow you to inspect the ballot
   it cast for you?

   YES, I would trust it   NO, I not would trust it

3) Would you trust a computerized voting system if did not allow you
   to monitor how it worked, but it did allow you to inspect the ballot
   it cast for you?

   YES, I would trust it   NO, I not would trust it

4) Would you trust a computerized voting system if it allowed you to
   monitor how it worked and allowed you to inspect the ballot it
   cast for you?

   YES, I would trust it   NO, I not would trust it

[Presumably Kurt will share the results with us. A
sequence of four answers (YES or NO) will suffice. PGN]

Search RISKS using swish-e

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 34

Saturday, 9 August 1986

Contents

- Non-Flying Airplanes and Flying Glass
  Jim Horning
- Failure Recovery, Simulations, and Reality
  Danny Cohen
- Ottawa Power Failure
  Dan Craigen
- Liability for Software Problems
  Peter G. Neumann
- Ozone hole
  Hal Perkins
- Re: Survey of Trust in Election Computers
  Chris Hibbert
- Nondelivery of RISKS-2.38 (8 April 1986) and other mail
  Communications Satellite [and PGN]
- Info on RISKS (comp.risks)

Non-Flying Airplanes and Flying Glass

Jim Horning <horning@src.DEC.COM>
Fri, 8 Aug 86 14:45:04 pdt

A number of people sent me information about the myth that the design flaw in the Electras wasn't caught because of an undetected overflow. (The most detailed information came from someone who wishes to remain anonymous.) Putting it all together, I am now convinced that the problem was not undetected overflow. Rather, it was a failure to simulate a dynamic effect (gyroscopic coupling) that had never been significant in piston-engined planes. So another myth bites the dust. But the true story should remind us that simulations are only as good as the assumptions on which they are based.

I solicit similar clarification of the story of the (then) new John Hancock Building in Boston (the one that resonated and shed many of its exterior glass panes when the wind came from a certain direction). I know that there was litigation about who was responsible for the additional costs: replacing the glass; installing a huge lead deadweight mounted on shock absorbers in an upper story to damp the oscillation; etc. I don't recall the final outcome. I do remember reading that there was a very narrow range of wind directions that would excite the resonance, and that the simulations of the design had unluckily missed that range. Maybe some readers of Risks know the details? Has there been a book or magazine article that explored the computer angle (if indeed there is one)?

Jim H.

Failure Recovery, Simulations, and Reality

<COHEN@B.ISI.EDU>
8 Aug 1986 18:38:58 PDT

In RISKS-3.27 Stephen Little, Computing & Information Studies, of Griffith Uni, Qld, Australia. reported that:

> I have been told of one major accident in which the pilot followed the drill for a specific failure, as practiced on the simulator, only to crash because a critical common-mode feature of the system was neither understood, or incorporated in the simulation.

Being a pilot I find this report most important and interesting.

I am sure that the readers of RISKS would be better served by having evidence to support such reports. Major (and responsible) newspapers have a verification procedures. Since RISKS cannot afford this I'd be delighted to help this process.

The best way to verify such a report is by a reference to the official accident investigation report. I'd be delighted to pursue this reference myself if anyone can give me details like the date (approximately), place (country, for example), or the make and type of the aircraft.

This is a plea to provide me with this information.

Danny Cohen.

[This is a very nice offer, and I hope someone can provide enough details to take you up on it! PGN]

Date: Sat 9 Aug 86 14:47:36-CDT
From: Dan Craigen <CMP.CRAIGEN@R20.UTEXAS.EDU>
Subject: Ottawa Power Failure
To: risks@CSL.SRI.COM

A brief fire at Ottawa Hydro's Slater Street station on the morning of

August 7th resulted in a loss of power to a substantial section of the downtown core. Even after 48 hours of effort, sections of the city were still without power.

[From the Ottawa Citizen (Friday, 8 August 1986)]

Top officials from Ontario and Ottawa Hydro today [Friday] are re-examining long accepted system reliability standards...
Ottawa Hydro engineering manager Gordon Donaldson said ``the system is built to be 99.99 per cent reliable ... now we will be looking at going to another standard of reliability -- 99.999 per cent.''
He also said that the cost would be huge -- many times the $10 million cost of the Slater Street station -- and hydro customers may not be prepared to accept the cost. ...
The Slater station is the biggest and was considered the most reliable of the 12 across the city. It has three units, each of which is capable of carrying the whole system in an emergency.
But ... all three were knocked out. ...
The culprit, an Ontario Hydro board [called a ``soupy board''] which monitors the equipment at the substation, didn't even have anything directly to do with providing power to the thousands of people who work and live in the area.
... its job is to make the system safer, cheaper and more reliable....
The board is considered so reliable that it doesn't have its own backup equipment. [!]

The economic costs of the power failure are expected to be in the millions of dollars. It is unlikely that the Ottawa birthrate will increase. As columnist Charles Lynch noted: ``The Ottawa power failure took place during the breakfast hour, not normally a time when Ottawans are being polite to one another, let alone intimate.''

We, at I.P. Sharp (Ottawa), lost both our VAXs; I have been unable to get onto Tymnet for the past two days; ATMs as far as a 100 miles distant from Ottawa were knocked out of commission -- the central computer that controls them is in the area of outage; Many traffic signals are still out; and a number of businesses still shut.

Dan Craigen

[Add this to the growing collection of problems in which a redundant system failed because of a weakest link in the redundancy itself! PGN]

Liability for Software Problems

Peter G. Neumann <Neumann@CSL.SRI.COM>
Sat 9 Aug 86 11:48:40-PDT

All week long I have been waiting for either someone else to submit it or for me to have a few spare moments to enter it: an item from the Wall Street Journal of last Monday, 4 August 1986, "Can Software Firms Be Held Responsible When a Program Makes a Costly Error", by Hank Gilman and William M. Bulkeley. A few excerpts are in order.

Early last year, James A. Cummings Inc. used a personal computer to prepare a construction bid for a Miami office-building complex. But soon after the bid was accepted, the Fort Lauderdale firm realized that its price didn't include $254,000 for general costs. Cummings blamed the error on the program it had used, and last October filed suit in federal court in Miami against the software maker, Lotus Development Corp. The suit, which seeks $254,000 in damages, contends that Lotus' "Symphony" business program didn't properly add the general expenses, resulting in a loss in completing the contract.

Lotus, based in Cambridge, Mass., disputes that contention, araguing that Cummings made the error. The case, however, has had a chilling effect on the software industry. For the first time, industry officials say, a case ma go to court that could determine if makers of software for personal computers are liable for damages when the software fails. Some software makers also worry that such a case, regardless of the outcome, may lead to other suits by disgruntled consumers. [...]

Software makers are particularly concerned about paying for damages resulting from faulty software -- rather than just replacing the software. Such "consequential" damages have been awarded in suits involving larger computers. Other types of damages from computer disputes "come from saying what benefits you were supposed to get compared with what benefits you didn't get," says Richard Perez, an Orinda, Calif., lawyer. Mr. Perez won a $2.3 million judgment against NCR Corp. for Glovatorium, Inc., a dry cleaner that said its computers didn't work as promised.

The article goes on to note that most PC software comes on an "as-is" basis, which doesn't provide for correction of errors. Under the limited warranties, the buyer does not even "own" the program. Illinois and Louisiana have passed "shrink-wrap" laws which imply that when you open the package, that is equivalent to signing a contract that lacks guarantee and prevents copying.

In the case of Cummings, they noticed they had left out the general costs, and added them as the top line of a column of figures. The new entry showed on the screen, but was not included in the total. Keep your eyes open for whether the blame is placed on a naive user not following his instructions, or on the software not doing what it was supposed to (or both).

Ozone hole

Hal Perkins <hal@gvax.cs.cornell.edu>
Fri, 8 Aug 86 03:17:48 EDT

In response to PGN's request for sources on the ozone hole...

The New York Time's Science Times section on July 29, 1986 had a long story on this (it starts on page C1). The gist of the story is that there's a big hole in the ozone layer over the south pole, nobody knows how it got there, nobody knows what it means, it could be a very serious problem, and scientists are investigating the situation.

As for computers and such, here are a couple of relevant paragraphs:

"The initial report of the hole by British scientists in March 1985 caused little excitement, partly because the British team in Antarctica was not well known among atmospheric scientists. Also, since their data came from ground instruments measuring the ozone in a direct line upward, they did not show the extent of the hole.

"But later last year, scientists at the National Aeronautics and Space Administration produced satellite data confirming the British findings and showing how big the hole was. NASA scientists found that the depletion of ozone was so severe that the computer analyzing the data had been suppressing it, having been programmed to assume that deviations so extreme must be errors. The scientists had to go back and reprocess the data going back to 1979."

Re: Survey of Trust in Election Computers

<Hibbert.pa@Xerox.COM>
Fri, 8 Aug 86 10:30:03 PDT

I'm afraid your questions are too vague for me to give yes or no answers. (I hope you'll give a count of non-respondents when you tell us how many YESes and NOs you got.) I'm not at all sure what it would mean for a voting system to allow me to monitor how it worked. Would it print out a trace of its execution? Would it let me know the running total of votes it had collected?

What would it mean for the system to allow me to inspect the ballot it cast for me? Does that mean the "computerized" aspect is merely a printer for ballots that will be counted later by hand or some other computer? Or does that mean that before I accept my votes it displays a summary for me to approve, and it then adds them into its running total?

I'm not convinced I would ever trust a system that only kept running tallys in software. If there aren't paper ballots printed, then there is no way to recheck the results. In this situation, the machine that later counts the paper ballots is much more important, and your questions don't address this part of the process.

Chris
 [We await Kurt Hyde's results...]

[Nondelivery of RISKS-2.38 (8 April 1986) and other mail]

Communications Satellite <COMSAT@MC.LCS.MIT.EDU>
Fri, 8 Aug 86 19:43:54 EDT

============ A copy of your message is being returned, because: ============

"HEWITT-RISKS" at MC.LCS.MIT.EDU is an unknown recipient.
============ Failed message follows: ============
Received: from MX.LCS.MIT.EDU by MC.LCS.MIT.EDU via Chaosnet; 8 AUG 86 19:42:12 EDT
Date: Tue 8 Apr 86 21:15:55-PST
From: RISKS FORUM (Peter G. Neumann, Coordinator) <RISKS@SRI-CSL.ARPA>

[REST OF MESSAGE TRUNCATED...]

[For the past week or so, I have been getting sequential notices of undeliverable mail from "Communications Satellite" -- four months after the original mailings of RISKS, and just another risk of running a forum.

There was a news item last week about an entire bag of US mail from aboard the Liberty Ship Caleb Strong from World War II (May 1944) that was just found undelivered by an exterminator in an attic in North Carolina. The Postal Service is trying to find the addressees, but was quick to add that it did not happen on their shift! (It blamed a soldier, who has since died.) Here are two related items that I just happen to have filed away.

Herb Caen's SF Chron column of 18 December 1973 noted a 1940 calendar mailed in 1939 to a customer in Utah that was returned "Addressee Unknown" during that week in 1973.

The Martha's Vineyard Gazette of 30 March 1973 noted a postcard mailed in Asbury Park NJ, postmarked 11 August 1914, addressed to West Summit NJ and forwarded to Edgartown, Mass. It arrived at that post office on 26 March 1973.

With sleet and snow and dark of night, now computers are doing it, too -- and they don't even need to find excuses. PGN]

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 2: Issue 38

Wednesday, 9 Apr 1986

Contents

- The UK Driving Vehicle Licensing Centre
  Brian Randell
- Computer crime wave
  Chris Hibbert
- Programming productivity
  Herb Lin
- Request for information about military battle software
  Scott E. Preece
- Aviation Week Technical Survey: AI & Aviation
  Werner Uhrig
- Info on RISKS (comp.risks)

The UK Driving Vehicle Licensing Centre

Brian Randell <brian%cheviot.newcastle.ac.uk@cs.ucl.ac.uk>
Tue, 8 Apr 86 12:03:45 gmt

Several newspapers and magazines here have carried stories about
the alleged activities of hackers regarding the Driving Vehicle Licensing
Centre - a very large computer system that has received much bad
publicity in the press and in parliament over the years because
of cost over-runs and delays.
Here is a sample, from the April 1986 glossy journal "Business":

"Computer hackers have been running a brisk racket "cleaning up" the
driving licences of wealthy business men. For a charge of [pounds] 100
a point endorsements have been erased from the files of the British
Government's Licensing Centre at Swansea and its supposedly impenetrable
computer ordered to issue new licences. Drivers who accumulate 12 penalty
points within 3 years are liable to ban or disqualifications. Reckless
driving, for instance, attracts 10 points; failing to stop after an accident
5.9 points; drunken driving 10 points (plus a 12 months disqualification).
Drivers' records at Swansea are held on the Department of Transport's
3081 Model G mainframe, whose manufacturers, of course, are not responsible

for its customers security procedures. About a year ago, an access code
number appeared on at least four "bulletin boards" - informal computer
games and information exchange facilities set up and used by home computer
enthusiasts (not in this instance mischevious schoolboys).
"I am not suggesting the number on the board was that of the DVLC", says a
source, "but it gave you access to a database with levels of password
protection. It was obviously a secure system and was related to DVLC
because the name headed the file. The access was not very privileged
but knowing the procedures allowed priority in the system and enabled you
to eliminate endorsements and order new licences to be issued."
Amendments to the DVLC mainframe were automatically carried through to
the back-up records kept on magnetic disc storage."

Such stories have inspired denials from the DVLC - for example in Datalink:

"The Driving and Vehicle Licensing Centre in Swansea has denied press
reports that computer hackers have broken into its database and wiped
traffic offenses off driver records.
The DVLC, which employs 1500 staff in a computer centre running a variety of
kit including two IBM 3083s, is adamant that its system is secure from
outside interference. "We have no dial-in facility, there's no electronic
access at all from off-site," a spokesman said.

Some 160 programmers work at the DVLC, and the spokesman admitted that
officials are "looking at internal arrangements" to see whether files have
been amended in return for payment."

My cynical view is that from most other sources such a denial would be
immediately accepted, and indeed it may well be true. However the thought that
such record tampering just might be going on, and so allowing banned drivers
back onto the roads, is a worrying one.

Cheers, Brian Randell - Computing Laboratory, University of Newcastle upon Tyne

ARPA : brian%cheviot.newcastle@ucl-cs.arpa
UUCP : <UK>!ukc!cheviot!brian
JANET : brian@uk.ac.newcastle.cheviot

computer crime wave

<Hibbert.pa@Xerox.COM>
Wed, 2 Apr 86 10:53:29 PST

There was an article in the March 31, 1986 edition of the Washington
Post's National Weekly Edition titled "The Computer Crime 'Wave': It's
more politician's bark than our byte".

After an initial few paragraphs in which the writer reminded us that
"national commissions that are set up to study and report on This Trend
or That Issue always end up concluding that the trend/issue in question
is a bigger national problem than anybody ever imagined", the article
reported on the "First Annual Statistical report" from the National

Center on Computer Crime.

"Over a two year period, the national center surveyed 130 prosecutor's offices in 38 states and asked how many computer crimes each office had encountered. ... The national center's survey of prosecutors came up with a grand total of 75 reported 'computer crimes.' Even that minuscule number, it must be noted includes some infractions that can only be classified 'computer crime' if you stretch the language considerably. One reported case involves ... a county prosecutor ... who got a friend in the motor vehicle department to delete two speeding tickets from his driving record. This is labeled 'computer crime' because the record was on a computer tape...

In short, this first national census says that 'computer crime,' by any stretch of the definition, is a statistically minute phenomenon. The antics of a few hackers have garnered grossly disproportionate attention from the media and the law-enforcement community. So-called 'computer crime' is novel and exciting, so it's hardly surprising that even a few cases would attract considerable notice.

But Legislators around the country are acting as if there really is a 'computer crime' problem. The center's study shows that 22 states passed new 'computer crime' legislation in the past two years. ..."

Chris

Programming productivity

<LIN@XX.LCS.MIT.EDU>
Sun, 6 Apr 1986 23:45 EST

> From: ihnp4!utzoo!henry at seismo.CSS.GOV
>
> I went and re-read Terry Winograd's old "Reactive Engine" paper. He comments, roughly: "If, by decree of God or ARPA, we were only allowed to run one user at a time on the PDP-10, just think of all the effort that would be invested in making that one user's time productive." Despite the enormous increases in computing power available to individual users since then, that has not happened: much of that extra power is simply being thrown away.

True enough. But why do you think that large amounts of effort invested would necessarily improve productivity? Despite long practice, for example, people can hold only a few ideas simultaneously in short term memory. There are mnemonic aids available, but they don't enable someone to do hundreds of times better.

I use this analogy because there is some evidence that limitations on short-term memory account for a variety of cognitive limitations, among which may be programming. Ultimately, it may the limitations of the human mind that prevent us from forever expanding our achievements.

> How many programmers, even ones working on life-critical software like

airliner flight control or fiercely difficult problems like
ballistic-missile defence, have the kinds of electronic and human
support that these thoughts suggest are possible?

That's easy. Not many. Indeed, military software procurement is by all accounts an utter mess.

Request for information about military battle software

Scott E. Preece <preece%ccvaxa@gswd-vms>
Mon, 7 Apr 86 09:43:05 cst

> [Parnas, quoted by Dave Benson]

> The other members of the SDI advisory panel that David Parnas was on
> and other public figures have said "Why are you so pessimistic? You
> don't have any hard figures to back up your claims." Parnas agreed
> that he didn't have any until he thought of the only one that he
> needed: ZERO. ZERO is the number of real systems that were trustworthy
> at first use. ZERO is the number of real systems that met unknown
> requirements at first use. ZERO is the number of prototyped systems
> that worked at first use. ZERO is the number of simulated systems that
> worked at first use. ZERO!

There are two essential, undefined terms in this statement: "first use" and "worked". The shuttle Enterprise, for instance, worked the first time they dropped it from its carrier 747. Was that its "first use", or do you count the many hours of simulation preceding that first flight? I wasn't there and have no idea whether there were bugs that showed up, but they clearly didn't keep the test from succeeding. Is that "working"?

The trouble with a debate like this is that it tends to force people more and more into idiotic dichotoomized positions. SDI software would obviously be a huge challenge to produce and validate. I have no hope it would work perfectly the first time used; I have no reason to believe it wouldn't work partially the first time it was used. The question of how perfectly it has to work is the central one. All the reports I've seen on both sides, including Parnas's essays, are hand waving. The task is too ill defined to be making statements about whether it can be done. The debate is silly. If you build the thing, you don't trust your security to it until you have been damned well convinced that it works; I am unwilling to accept the statement that "You can never be convinced that it works," when daily we all trust our lives dozens of times to things that we have been convinced work. There are plenty of good and, I think sufficient, arguments for not building SDI without claiming that it can't be done.

--
scott preece
gould/csd - urbana
ihnp4!uiucdcs!ccvaxa!preece

Aviation Week Technical Survey: AI & Aviation

Werner Uhrig <CMP.WERNER@R20.UTEXAS.EDU>
Tue 8 Apr 86 11:06:41-CST

[I am sure, readers of AVIATION and RISKS are interested also;
for somewhat different reasons, of course ....   ---Werner]

Date: Wed 26 Mar 86 09:08:28-PST
From: Oscar Firschein <FIRSCHEIN@SRI-IU.ARPA>
Subject: Aviation Week Technical Survey

AILIST readers might be interested in the following:

Aviation Week and Space Technology, Feb. 17, 1986 has a technical survey of artificial intelligence, mostly applied to military applications. Included are the DARPA-supported programs in Pilot's Associate and the Autonomous Land Vehicle (ALV) and the VLSI lisp machine being built by Texas Instruments.

Company profiles include McDonnell Aircraft's work in the Pilot's Associate and avionics maintenance expert system; Boeing's AI Center; MITRE's work in natural language understanding; Grumman's decision support systems; Hughes AI center; and Westinghouse avionics troubleshooting expert system.

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 35

Monday, 11 August 1986

Contents

- Flying windows on the Hancock Building
  Remy Malan
- Pilots and counter-intuitive maneuvers
  Martin Minow
- Mail adrift
  Mike McLaughlin
- Laserprinter dangers
  Niall Mansfield
- A bit of humor and even philosophy
  Willis Ware
- Official Report on Chernobyl disaster
  Robert Stroud
- Info on RISKS (comp.risks)

Flying windows on the Hancock Building

Remy Malan <remym%tekig5.tek.csnet@CSNET-RELAY.ARPA>
Sun, 10 Aug 86 08:37:32 PDT

While at school in Cambridge, MA. I took a course in decision analysis.
One of the examples given in class was the case of the Hancock Building.
This is how I remember it:

A model of the Hancock Building and the surrounding structures was tested in a wind tunnel. The wind direction in the initial tests was incremented by 45 degree intervals. The model behaved well for these tests. Later, after the problem occurred on the real structure, more testing [at a finer mesh] revealed very narrow bands in wind direction in which resonance did occur. The 45 degree increments were too coarse to pick out the resonant zones.

(I believe that their initial tests were done informally, as a kind of favour, and so were not very rigourous.)

*This is all from memory, so my apologies if I didn't get it quite right.

A. Remy Malan

Pilots and counter-intuitive maneuvers

Martin Minow, DECtalk Engineering ML3-1/U47 223-9922 <minow%rex.DEC@decwrl.DEC.COM>
10-Aug-1986 0025

This is from memory, and it's late, so bear with me:

A very recent Smithsonian (June 86?) had an article on flight simulators --
the same month as the Scientific American article. In it, the chief instructor
for one of the airlines related that, a few months ago, he flew as the
flight engineer on a commercial flight. The plane encountered a wind-shear
situation on take off. The instructor, from his flight engineer's position,
reminded the pilot that the correct recovery for wind-shear is opposite to
the correct recovery for a stall (which has a similar appearance to the pilot).

Hope this reassures your pilot subscribers. By the way, accident investigation
reports are usually summarized in Aviation Week and Space Technology.

Martin Minow
minow%rex.dec@decwrl.dec.com

---

mail adrift

Mike McLaughlin <mikemcl@nrl-csr>
Sun, 10 Aug 86 11:24:12 edt

Personal item, no documentation known: I once purchased a used USPS station
wagon at GSA auction for $350. While cleaning it out, my wife and I found
well over a hundred pieces of undelivered mail. We trashed all but the
first class - and dropped 30 or 40 pieces into the nearest mail box. Some
were over five (5) years old. We watched the paper for days, but saw no
items about late mail.

Only relevance to RISKS is that people will _always_ be imperfect.

- Mike

[And how often do we assume that a system will work
properly in the face of that statement?! PGN]

Laserprinter dangers

Niall Mansfield <MANSFIEL%DHDEMBL5.BITNET@WISCVM.ARPA>
Mon 11 Aug 86 18:29:51 N

>From: Graeme Hirst <gh%ai.toronto.edu@CSNET-RELAY.ARPA>
>Subject: Re: Laserprinter dangers

Sean Malloy dealt with the ease of forging with laser printers. A more general point is that forging ANY computer-produced item, be it a hard-copy output or a message on a wire, is easier than forging old-style pieces of paper, etc., because:-

1. The machinery involved is cheap - bytes on a wire which have come from a cheapo toy computer just look just like expensive DEC or IBM bytes. (Coiners need expensive metal presses)

2. You can realistically attain a 100% perfect forgery - my bogus bytes look just the same as real ones. (Coiners presumably have difficulty making the right alloys, but worse, have to copy the shapes on the coin - how do they know when their product is "good enough"?)

3. The skills required are, more or less, the same for producing ordinary software as for producing forgeries - software is software, whether legal or otherwise. (It is also true that an engraver uses his same skills whether he is forging banknotes or producing a bookplate; the big difference however, is in the widespread distribution of skills needed for forging - there are very few qualified engravers, but lots of "qualified" programmers).

In summary, a lot of people are finding themselves in a position they were never in before - not only have they all the skills and equipment necessary for a particular type of crime, but increasingly they are being presented with opportunities to commit those same crimes. Ergo ...

A bit of humor and even philosophy

<willis@rand-unix.ARPA>
Mon, 11 Aug 86 16:07:38 PDT

In the Washington Post, July 30 1986, pg A-23, columnist James J. Kilpatrick discusses the nomination and confirmation of Daniel Manion as appellate judge. He laments at length the lack of support for the individual, notes that a keen sense of justice is not all that important for appellate judges anyway if they have a good knowledge of the structure of law which is what they really rule on. He goes on to note that the analysis of pertinent law and the detailed writing will likely be turned over to law clerks anyway.

The last paragraph of the article is the clincher and source of humor.

> "In sum, I fear not for the republic, or for the 7th Circuit, when Manion joins the club. Give him an intelligent clerk and a good word processor, and the gentleman may look forward to many happy years on the bench."

Do you suppose it could be called an application of AI, when software offsets presumed deficiencies of appointed officials?

Are things such as this off-the-cuff suggestion an early step of having software front for the performance and/or the beavhior of public officials? And with what unseen, possibly unknowable, risks?

Official Report on Chernobyl disaster

Robert Stroud <robert%kelpie.newcastle.ac.uk@Cs.Ucl.AC.UK>
Mon, 11 Aug 86 15:01:08 bst

The following article appeared in yesterday's Observer, and is reproduced here without permission:

Robert Stroud,
Computing Laboratory,
University of Newcastle upon Tyne.

ARPA robert%cheviot.newcastle@ucl-cs.ARPA
UUCP ...!cheviot!robert

===

"Chernobyl report blames turbines" p.6 Observer, Sunday 10 August 1986

by Robin McKie and Laura Veltman

Soviet operators who experimented with turbines and alternators at the Chernobyl plant are to be blamed for the nuclear disaster there.

Western experts who have recently visited Chernobyl say that the full Soviet accident report which is expected to be published this week, will blame 'human error' and 'misuse' of turbines for the chain of events that led to the disaster in April.

But many believe the explanation is inadequate and that it is being promoted mainly to protect the country's nuclear construction programme.

'The theory moves the source of the accident from the reactor itself to the turbines which are housed separately,' said Mr Peter Potter, a British nuclear expert who has seen many Soviet reactors.

'By maintaining that human error and turbine problems were really to blame, the Russians could say that their reactors have no serious design flaws. They could then avoid calls for closures of other reactors or for the implementation of drastic redesign work.'

The Soviet theory argues that the Chernobyl accident was caused by a total loss of electricity supply to the pumps which circulate cooling water through the heated reactor core. One Western scientist, Professor Leslie Kemeny, of the University of New South Wales' nuclear engineering group, does believe that an accident with the electricity-generating turbines - which are worked by steam heated in the reactor - triggered the disaster.

Prof Kemeny, who took detailed samples of air, water and soil contamination during a recent visit to the Chernobyl area, said: 'The loss of electricity to the pumps was due to human error. During the night of 25 April, the turbo-alternator linked to Reactor 4 at Chernobyl was undergoing a "run-down" experiment. In effect, this meant that engineers were studying the behaviour of the turbines while they were being run down. Throughout the hour of the experiment, alternative energy sources should have supplied replacement power for the pumps. But this did not function, and the reactor was left uncooled.'

Normally, the reactor's own electricity should have been used to run the cooling pumps. During a run-down, an alternative source should have been switched on automatically. It was this which failed at Chernobyl. Without cooling water, the reactor's temperature was sent soaring - with dire effects on its uranium fuel, zirconium cladding and graphite core.

First the remaining water inside the reactor heated up, forming steam which began to react with the zirconium to produce hydrogen. The pressure of the steam and the hydrogen eventually cracked the reactor core's outer tube. Finally, when air mixed with the hydrogen, it exploded and set fire to the graphite in the core. The result was an inferno which sent radioactive debris puring over much of Europe.

Despite his support for the accident theory, Prof Kemeny criticised the Russians for failing to build pressure domes over the reactor core. 'I stand by my belief that the Chernobyl reactor was safety-deficient,' he said. 'American, German, French and British reactors have pressure vessels and strongly reinforced concrete structures to contain such radiation releases.'

But other nuclear experts cast doubt on the turbine theory. 'I don't think it is the whole story,' Mr Potter said. 'The explanation begs some questions. Why didn't the alternative back-up power supples switch on automatically, and what caused the power surge which the Russians say occurred at the time of the accident? I think there was another factor - concerned with the reactor itself - which was involved but which the Russians do not want highlighted for political reasons. They would find it very inconvenient if it was shown that there were serious generic design faults in all their RBMK reactors, the ones like the Chernobyl reactor. They are not going to let that idea spread'

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 36

Tuesday, 12 August 1986

Contents

- Another Medical Risk?
  - Lee Breisacher
- RISKy Business in Surgery
  - Mark Jackson
- Reliance on word-processors discussed in the Israeli Supreme
  - Ady Wiernik
- Expert Systems - The New Cop on the Beat
  - Laws via Fred Ostapik
- Chernobyl
  - Art Evans
  - Dick Karpinski
- Air Traffic Control computer failure
  - Dan Melson
- Possible failures of BMD software
  - Herb Lin
- A note about stories "from memory"
  - Henry Mensch
- Info on RISKS (comp.risks)

Another Medical Risk?

<Breisacher.OsbuSouth@Xerox.COM>
12 Aug 86 09:25:49 PDT (Tuesday)

From the August PSA Airline magazine, extracted from an article about inventors:

[There's a photo of Dr. Kwoh in surgical garb in an operating room leaning over a dummy patient with some elaborate equipment surrounding its head. The caption reads:]

Robotic surgery is a reality because of the obsessive work of Yik San Kwoh, medical research and development director of Long Beach Memorial Hospital. His computer controlled "surgeon," capable of conducting brain surgery within an accuracy of 1/2000 of an inch, was the result of three years of incessant programming.

[From the text of the article:]

Yik San Kwoh, medical research and development director of Long Beach Memorial Hospital, explains, "I've got two Apple computers at home and three IBMs. I spend so much time on those damn things that I get sick of it. Only then can I stop."

It took three years of programming and reprogramming for Kwoh to turn and industrial robot into a surgical instrument capable of conducting brain surgery.

[As usual, we must weigh the risks of using such equipment against the risks of NOT using it. On the other hand, the description makes it sound like he programmed this thing the way I wrote my first couple programs (FORTRAN in the early 70's) -- dive in and start writing code then keep debugging til it sorta works.]

Lee

RISKy Business in Surgery

<MJackson.Wbst@Xerox.COM>
12 Aug 86 07:56:26 EDT (Tuesday)

From /Programmers at Work (1st Series): Interviews/, by Susan Lammers (Microsoft Press, 1986):

"My most amazing experience, though, was a phone call I got right after I started Iris, from a surgeon who was using Symphony for real-time data analysis during open heart surgery. It is sobering to think that someone was lying on an operating table potentially relying upon my program running properly. It reminds one of the real responsibility to the end users."

-- Ray Ozzie
project leader for Symphony

Reliance on word-processors discussed in the Israeli Supreme Court

Ady Wiernik <ady%taurus.BITNET@WISCVM.ARPA>
Tue, 12 Aug 86 21:19:31 -0300

Rules of Court in Israel fix a time limit for bringing an appeal to the Supreme Court against a decision of an inferior Court.

A lawyer applied to Supreme Court for an extension of the period to appeal. He has missed the statutory period by two days. His excuse was that the word-processor in his office (that has been recently installed) malfunctioned.

The text of the appeal that was typed into the computer has been erased because of that computer malfunction. He called the maintenance personnel. They promised that the malfunction would be shortly repaired, but actually, it lasted longer, causing him not to be able to bring the appeal at the same day.

The appellant claimed that the trouble with the computer was an "act of god", Force Majeure, which is considered a special ground that entitles him the desired extension.

The court has rejected this argument.

In his judgement, Registra Tzur of the Supreme Court said: "Indeed, the computer is very useful, but one must prepare for possible malfunctions in its operation. When there is no computer, the good old typewriter should replace it."

This decision is the first recorded judicial reference to the use of word-processing devices in lawyer offices, and displays the dangerous results of reliance on high-tech.

Ady Wiernik

Expert Systems - The New Cop on the Beat

<Laws@SRI-STRIPE.ARPA [courtesy of Fred Ostapik]>
Mon 4 Aug 86 22:38:23-PDT

The FBI has developed Big Floyd, an expert system to assist in criminal investigations. Similar programs are being developed to catch drug smugglers and target potential terrorists. The EPA wants to identify polluters; the Treasury Department is looking for money-laundering banks; the Energy Department would like to find contractors who cut corners; the Customs service is after drug smugglers; the IRS is developing a system to spot tax cheaters; the Secret Service is working on a classified system to point out potential presidential assassins; and the FBI's National Center for the Analysis of Violent Crimes is developing expert systems to identify potential serial killers, arsonists, and rapists. Systems to target counterfeiters and bombers are also being built. -- Michael Schrage, The Washington Post National Weekly Edition, Vol. 3, No. 40, August 4, 1986, p. 6.

Chernobyl

"Art Evans" <Evans@TL-20B.ARPA>
Tue 12 Aug 86 11:34:21-EDT

In RISKS-3.35, Robert Stroud comments on "Official Report on Chernobyl disaster". Although the discussion of what actually triggered that disaster is interesting, I choose to focus instead on how the Russian explanation was interpreted by others (not by Mr Stroud).

Quoting from the post:

> But many believe the explanation [offered by the Russians] is inadequate and that it is being promoted mainly to protect the country's nuclear construction programme.

No justification is given for this belief. A Peter Potter is quoted as saying

> By maintaining that human error and turbine problems were really to blame, the Russians could say that their reactors have no serious design flaws. They could then avoid calls for closures of other reactors or for the implementation of drastic redesign work.

This claim may in fact be true, but we are given no evidence.

Note what is happening: The Russians offer a technical explanation for the disaster. A western nuclear expert says the explanation is inaccurate and was offered for political reasons. But, no reason other than political is given for this skepticism. The Russians may well be lying, and if there is evidence I would like to see it. Lacking such evidence, though, the public would be better served by less misleading pronouncements by "experts".

Art Evans

Chernobyl

Dick Karpinski <dick@cca.ucsf.edu>
Tue, 12 Aug 86 11:13:17 PDT

The only unadvertised design deficiency that I know of in the Chernobyl reactor is that it has a positive coeficient of reactivity with respect to temperature. That is, when the temperature goes up, so does the rate of nuclear fission. Such a design would be ruled out here, claims my source, a former reactor containment vessel engineer. Surely, such a design would make the sort of accident which occurred more likely.

Dick

Dick Karpinski Manager of Unix Services, UCSF Computer Center
UUCP: ...!ucbvax!ucsfcgl!cca.ucsf!dick (415) 666-4529 (12-7)
BITNET: dick@ucsfcca Compuserve: 70215,1277 Telemail: RKarpinski
USPS: U-76 UCSF, San Francisco, CA 94143

Air Traffic Control failure

Dan Melson <crash!pnet01!dm@nosc.ARPA>
Mon, 11 Aug 86 23:47:21 PDT

Computer failures at Air Route Centers are not as uncommon as we'd like, but they're not as nasty as they could be. Despite the fact that the computers currently used are more than fifteen years old, they seem to handle the load well enough for the present. When the primary computers (IBM 9020's) go down, however, the DARC backup system does not furnish the controllers with nearly as much data, and it is far more difficult to get automated tasks done.

There is currently a new computer system in the works, and when it is operational, delays due to computer failure should dramatically decrease. The estimate for this is 'around 1990'.

At any rate, even the bachup systems are far more pleasant than doing all of the work manually.

DM

Possible failures of BMD software

<LIN@XX.LCS.MIT.EDU>
Tue, 12 Aug 1986 00:38 EDT

I'm working on a paper on potential software-induced difficulties and problems that might accompany the deployment of a BMD system. I'd like to enlist the collective imagination of the list on examples apropos to this paper.

Please constrain your imagination by the limits of the possible (e.g., it is impossible for an X-ray laser to shoot x-rays at ground targets, but it is not impossible that the firing of an X-ray laser creates an electromagnetic pulse that has unanticipated effects). Please specify the scenario in as much detail as you can. I am not specifying a system architecture, so please tell me the one(s) you have in mind in your scenario(s); that is necessary because softare -- by itself -- is harmless no matter how buggy it is. Also remember that BMD has significant capability against satellites.

Thanks. Acknowledgements will be provided if you so desire.

Herb Lin

A note about stories "from memory"

Henry Mensch <henry@ATHENA.MIT.EDU>
Mon, 11 Aug 86 23:44:12 -0500

I hate to sound like a nit-picker but I've noticed a rash of stories which begin with words like "If I remember correctly ..." or "It's pretty late, so expect errors." Is this sort of thing a product of having such powerful communications tools at our fingertips?

Once these things happen we seem to spend a lot of time saying "Well, *I* thought it went this way. . . " In discussing risks to the public, we risk wasting our time doing these tasks, which could be avoided with a bit of research.

Striving for better communications,

 -

Henry Mensch | Technical Writer | MIT/Project Athena
henry@athena.mit.edu ..!mit-eddie!mit-athena!henry

[On the one hand, it is nice to be precise. On the other hand, if the report is novel and interesting, perhaps RISKS provides a medium for getting feedback from an expert on a matter that would otherwise go unreported. But, I certainly appreciate it when contributors take a little time to track down the reference -- and especially when they cite that reference. PGN]

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 37

Thursday, 14 August 1986

Contents

- Computer Viruses
  - Robert Stroud
- On knowing how hard a system is to make work
  - Bob Estell
- COMSAT and the Nondelivery of Mail
  - Rob Austein
- Exploding Office Chairs
  - Jonathan Bowen
- Info on RISKS (comp.risks)

Computer Viruses

Robert Stroud <robert%kelpie.newcastle.ac.uk@Cs.Ucl.AC.UK>
Wed, 13 Aug 86 20:29:54 bst

Here is something I found in the Times yesterday. Since it is marked "Reuters" I assume it originated in the States so you may have seen it already. What is your policy on posting copyrighted articles? This is the entire text and I have not made any excerpts. On the other hand, I have acknowledged the copyright. There has been a fuss about this in net.unix recently, so I am rather concerned not to get myself, the University or you into trouble.

[RISKS is a non-profit educational operation. I believe that it is quite appropriate to quote an article under such circumstances -- with attribution. There is a burden on all of us to use it accordingly. PGN]

One of the "computer comics" (free journal made up of half news/features and half job adverts) called Datalink has a front page story about the X-ray machine in Texas killing a patient. I remember this coming up in RISKS some time ago, and you are quoted in the article as follows:

"Specialists in the field of software reliability have long been predicting fatalities caused by bugs. Peter Neumann of the US ACM claimed that the ACM's software engineering group had monitored 16 deaths caused by defective programs. "This is just the tip of the iceberg", he said.

[Actually I thought I mentioned to him that there were at least 16 CASES of computer-related deaths (a subsequent closer count by me shows that there are 24 different cases in my files). The total number of deaths in those cases is over 716. There were also three Soviet nuclear sub accidents with unknown tolls. PGN]

Manny Lehman is also quoted as being "not surprised - this is merely the front-runner of a thing we're going to see a lot of".

The same issue of Datalink also contains a story about how a problem with some new software led to rumours that Tetley's brewery had stopped production - while they were installing it, they ran into problems and to save time, tried to contact the programmer who was on holiday in Scotland. Somehow the messages got distorted en-route...

It's a nice anecdote but perhaps not really a RISK! However, I'll send it in if you're interested. People can take beer very seriously in the UK...

[Please send it! PGN]

==

Here is an article from yesterday's [London] Times (August 12th, "Computer Horizons"). Although it is couched in somewhat exaggerated tones(!), the consequences of failure are the same, whether induced by sinister bogeymen or simply design faults.

By coincidence, I recently came across a reference to the paper by F. Cohen of the University of Southern California entitled "Computer Viruses: Theory & Experiments", which apparently suggests that a Unix virus could gain root privileges within an hour, so maybe there is something to be worried about after all! [A few minutes is well within an hour... PGN]

Perhaps some of the "sources who spoke on condition they would not be identified" will read this and would like to comment further, (anonymously of course...)

Robert Stroud, Computing Laboratory, University of Newcastle upon Tyne.

ARPA robert%cheviot.newcastle@ucl-cs.ARPA, UUCP ...!ukc!cheviot!robert

==

"The 'virus' threat to defence secrets" (c) Times Newspapers Limited 1986

from Christopher Hanson in Washington

American Scientists are struggling to protect computer networks - vital in areas ranging from national defence to banking and air traffic control - against a potentially devastating weapon called a computer virus.

Computer security experts in the US government say the "virus" is a high technology equivalent of germ warfare: a destructive electronic code that could be inserted into a computer's program, possibly over a telephone line, by a secret agent, terrorist or white collar criminal.

When a computer virus attacks it wipes out crucial memory data or otherwise causes high technology equipment to behave erratically, according to sources who spoke on condition they would not be identified.

They said a computer virus attack might bring a major weapons system to a standstill, throw a computer-guided missile off course, or wipe out computer stored intelligence. "The government is concerned and we are pursuing solutions," one security official said.

Computer security experts have created experimental viruses in a bid to find defences, but there had been no breakthroughs.

Both the military's computer nets and the highly automated US banking system are vulnerable to "catastrophic collapse", according to a recent Georgetown University report by a group of government and private counter-terror experts. Urging that the pace of defensive research be quickened, it said the computer virus threat was "a matter of great concern...There do not appear to be any quick and easy defences or overall solutions to the problem."

As to the banking system, the report warned: "The four major electronic funds transfer networks alone carry the equivalent of the federal budget every two to four hours. These almost incomprehensible sums of money are processed solely between the memories of computers, using communications systems that are vulnerable to physical disruption and electronic tampering."

Computer viruses are designed to replicate themselves like a living organism, spreading throughout a computer netork, government scientists said. Viruses can spread from one computer system to another during electronic linkups and might lie undetected for months or years before going on the attack at a pre-determined time.

Before it begins to disrupt a system, a computer virus would be inconspicuous, containing only a few hundred "bytes" in a program that might total hundreds of thousands. Even the most carefully designed computer security barriers can be vulnerable, the Georgetown report said.

Another way the viruses could spread was through computer discs which computer users often copy and share. Scientists say the computer virus idea may have originated in a 1975 science fiction novel, "The Shockwave Rider". Intrigued computer buffs began tinkering and by the early 1980s had turned fiction into fact with experimental viruses. (Reuter)

On knowing how hard a system is to make work

<"SEFE::ESTELL" <estell%sefe.decnet@nwc-143b.ARPA> [or estell@nwc-143b]>
14 Aug 86 11:06:00 PST

I think there is a risk in solving computing problems too easily. A San Diego friend says that "The trouble with doing a project right the first time is that no one knows how hard it was." Though that happens infrequently, he's got a point. In most fields, accomplishment can be measured by effort, along with talent, luck, and some other things. The scholar who breezes through school often knows how hard it is, based on the hours spent in the library and the lab; the athlete whose graceful moves seem effortless knows how close to the limit she plays. But lots of "good" computing systems are joint ventures between a hardware designer of generic computer power, and a software designer of some particular algorithm; neither really knows how hard the machine works to solve a particular problem. Often it's only after the system fails that we realize that it was operating at its limit before we increased the load. That's in part because many programmers just write code, with little attention to thorough analysis & design as urged by Don Knuth's work; and in part because hardware designer and software end-user often never meet; and in part because the field is so broad and demanding that one person can't know it all.

There's another old saying, that an expert is someone who avoids all the minor errors on his way to the colossal blunder. That points up the risk of being so bright (or lucky?) that one never fails (or is even stressed) by routine assignments; and finally assumes a prominent role in a major, high risk program.

Maybe we should give some thought to having major computing projects headed by people who have reached their limits at least once along the way; not that they have failed, but that they have had to try again. [A winner is one who gets up one more time than he goes down.] With that in mind, does anyone know the "track record" of the leaders of some high risk projects; e.g., SDI? I'm sure these folks have impressive credentials; I just wonder if they've ever explored their own limits.

Bob

[Nondelivery of RISKS-2.38 (8 April 1986) and other mail]

Rob Austein <SRA@XX.LCS.MIT.EDU>
Thu, 14 Aug 1986 03:16 EDT

> Date: Friday, 8 August 1986 19:43-EDT
> From: Communications Satellite <COMSAT@MC.LCS.MIT.EDU>
> "[For the past week or so, I have been getting sequential notices of undeliverable mail from "Communications Satellite" -- four months after the original mailings of RISKS, ... PGN]"

COMSAT stopped being able to deliver messages of any serious length sometime around last December, and didn't really get fixed until mid-May (changing of the guard, had to scare up a new COMSAT hacker). During that time a couple of Really Dedicated People were faithfully saving all the messages that COMSAT was dropping on the floor. Ever since COMSAT was fixed these messages have been being dribbled back into the mail queue, 10 or 20 at a time (not practical to filter them, given the volume). The fact that it is now August and we still aren't done should give you some idea of the volume of mail that MC handles.

We announced this on Arpanet-BBoards (and other places) when we started dribbling the mail back in. Of course, that was a while ago....

--Rob

Exploding Office Chairs [A Peripheral Risk of Sitting Before a VDT?]

Jonathan Bowen <bowen%sevax.prg.oxford.ac.uk@Cs.Ucl.AC.UK>
Thu, 14 Aug 86 15:16:30 GMT

Below are extracts from two reports in the Guardian; the first rather jokey and the second less so, presumably after the journalist realised the seriousness of the problem.

Exploding chairs a pain in the office (Monday, 11th August 1986)

A new hazard at work, the exploding office chair, is facing - or, rather, the reverse - Britain's white collar workers. The problem is now under investigation so that up to 2 million minds, and a similar number of bottoms, may rest more easily. So far, 11 swivel chairs around the country are known to have gone off with a bang. In three cases the exploding chairs have caused injury, probably because the sitters have been sent sprawling as the bottom drops out of their world.

The problem has cropped up with adjustable office chairs fitted with nitrogen gas cylinders in place of the conventional springs in their height control mechanism. Preliminary findings suggest that metal fatigue cracks can develop in the cylinders, possibly caused by the poor chairs being asked to cope with more than they can bear.

Exploding chairs' two-year history (Tuesday, 12th August 1986)

The danger of office chairs exploding has not previously been made public because of official reluctance to raise an "alarmist scare," it emerged yesterday. The public has not been warned about blasts scattering stell fragments and metal bolts caused by failures in adjustable chairs fitted with nitrogen cylinders instead of conventional springs. Cases of serious injury came to light two years ago. ...

In September 1984, the Consumers' Association passed to the Health and Safety Executive (HSE) details reported by consumer organisations in Europe of incidents involving office chairs. They included accounts of two deaths, one in Belgium and the other in West Germany, where, it was reported, a piece of steel had penetrated a victim's brain through the eye. .... The HSE has stressed that only 11 incidents, three of which caused injury, are known to have occurred in Britain - where up to 2 million of the chairs are in use.

Has this story broken in the US yet? How many of you are sitting at your VDU on such a chair? This is the time to take a quick peek below you, and take appropriate defensive action if necessary. You have been warned!

Jonathan Bowen, Research Officer, Distributed Computing Software Project
Oxford University Computing Laboratory, Programming Research Group
8-11 Keble Road, Oxford OX1 3QD, England, Tel: +44-865-54141 x293
JANET: bowen@uk.ac.oxford.prg
UUCP: ...seismo!mcvax!ukc!ox-prg!bowen (bowen@ox-prg.uucp)

[Some persons talked into buying this chair
were evidently given a bum steer! PGN]

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 38

Sunday, 17 August 1986

Contents

- Computer gives away California state funds
  Rodney Hoffman
- High-Tech Sex Ring: Beware of Whose Database You Are In!
  Peter G. Neumann
- Computer Viruses
  Chris McDonald
  Paul Garnet
  Matt Bishop
- Computer Viruses and Air Traffic Control
  Dan Melson
- Re: Traffic lights in Austin
  Bill Davidsen
- Info on RISKS (comp.risks)

Computer gives away California state funds

<Hoffman.es@Xerox.COM>
15 Aug 86 13:51:39 PDT (Friday)

From the Los Angeles Times, August 15 1986, page 2:

> A computer error caused California's check-writing system to issue $4 million in interest-payment checks to bondholders who hold a type of bond on which no such payments were due. Deputy state Treasurer Liz Whitney explained that those bonds are of the "zero coupon" type, which are held for a period of years and redeemed with accumulated interest at maturity rather than bearing interest on a monthly or yearly basis. The treasurer's office learned of the error last Friday, she said, when a recipient inquired about the check's validity, and stop-payment orders were issued. By Wednesday, all but a few checks totaling $33,000 had been recovered.

No further details are given about the nature of the computer error.

-- Rodney Hoffman

High-Tech Sex Ring: Beware of Whose Database You Are In!

Peter G. Neumann <Neumann@CSL.SRI.COM>
Fri 15 Aug 86 19:37:38-PDT

From the San Francisco Chronicle, Friday 15 August 1986:

POLICE SAY ARRESTS IN MARIN SMASHED HIGH-TECH SEX RING
by Torri Minton and Katy Butler

A sophisticated prostitution ring that kept computerized records on more than 12,000 patrons has been broken after a three-month investigation, authorities in San Jose said yesterday. The ring, known as EE&L Enterprises, collected $3.5 million a year dispatching at least 117 prostitutes by electronic beeper to cities all over Northern California from a computerized command center in San Rafael, according to San Jose vice Lieutenant Joe Brockman. ``It's a top-class operation -- the largest prostitution ring, to our knowledge, in Northern California," Brockman said. He said that the business took in more than $25 million during the eight years it was in business...

Records seized by police ... included customers' names, telephone numbers, credit card numbers, sexual preferences and comments by the prostitutes... The office was equipped with four desks, several IBM computers, a photocopier, a paper shredder and a wall poster announcing that ``Reality is nothing but a collective hunch.''

On-line SuperCalifornication?

Computer Viruses

Chris McDonald SD <cmcdonal@wsmr06.arpa>
Fri, 15 Aug 86 7:47:01 MDT

[This is included because so many of you do
not seem to know the Cohen reference. PGN

Robert Stroud references a paper by Fred Cohen on "Computer Viruses." The full text of the paper can be found in several public souces. The most available for US readers is the minutes of the 7th DoD/NBS Computer Security Conference, Sept 24-26, 1984, pages 240-263. The paper is not exclusively concerned with any one particular operating system. It defines a "virus" as "a program that can infect other programs by modifying them to include a possibly evolved copy of itself." The paper references Ken Thompson's acceptance speech on the Turing Award, "Reflections on Trusting Trust," which was published in the August 1984 "Communications of the ACM." The reference, however, is only for purposes of illustrating what Fred proposes is a "limited" virus.

[That paper includes the wonderful C compiler Trojan horse lurking
in wait for the next recompilation of the UNIX LOGIN procedure. PGN]

A close reading of the paper would reveal that very specific factors have to exist for a "virus" to become "virulent." The most interesting facet of the paper is really the question it raises as to whether the Bell-LaPadula and the Biba models on mathematically defining "secure systems" even addresses the potential of a "virus" attack.

Computer Viruses

<pgarnet@nswc-wo.ARPA>
Fri, 15 Aug 86 12:14:22 edt

Another paper by Fred Cohen is "Recent Results in Computer Viruses", written while at Lehigh University. The copy I have does not have a date on it, but I believe it was written sometime around the spring of 1985.

Anybody else know of any good, technical papers on the subject?

Paul

Re: Computer Viruses

Matt Bishop <mab@riacs.ARPA>
Fri, 15 Aug 86 07:28:27 -0700

If anyone wants to read an interesting science fiction book about computer viruses (and things of that ilk) try reading John Brunner's "Shockwave Rider." Briefly, it's about a man who puts computer viruses into the worldwide data banks, enabling him to do all sorts of illegal things such as change identities. Quite interesting, at least from the viewpoint of computer security!

Matt Bishop

[I think we included mention of "Shockwave Rider" in RISK long ago.
However, with the interest in viruses and our large number of new
readers, I am not trying to avoid all duplication -- especially with
the distant past. PGN]

Computer Viruses and Air Traffic Control

Dan Melson <crash!pnet01!dm@nosc.ARPA>
Sat, 16 Aug 86 01:13:47 PDT

Those who fly regularly will be somewhat relieved to note that all terminals of the ARTS and NAS systems, except master consoles (and a few others hardwired straight into the machine and on site) are limited in what they can input, nor can they escape the ATC program. Furthermore, I am not aware of any means whereby employees can access any of the FAA's computers from other than known sites. This also explains why there are so few ATC's on any net, despite the large amount of computer work associated with the job today.

DM

[Beware of Trojan horses bearing gifts that look like sound programs, officially installed through proper channels. There is also the problem of accidental viruses such as the ARPANET collapse of 27 October 1980. (See Eric Rosen's fine article in the ACM Software Engineering Notes 6 1 Jan 81, for those of you who have not seen it before.) PGN]

Re: Traffic lights in Austin

<davidsen%kbsvax.tcpip@ge-crd.arpa>
15 Aug 86 10:57 EST

[From: Davidsen <davidsen%kbsvax@kbsvax.tcp-ip>]

I would call a 2% clean failure rate a success. If the two intersections had failed in an unsafe mode, such as green in both directions, it would not have been acceptable. If the lights had "stuck" showing green one way and red the other, it could have caused severe delays. For the light to cleanly go out is probably acceptable. Most drivers seeing a light with no signal showing will use adequate caution to prevent accidents.

-bill davidsen

```
 ihnp4!seismo!rochester!steinmetz!--\
                                     \
              unirot ------------->---> crdos1!davidsen
                   chinet ------/
      sixhub ---------------------/        (davidsen@ge-crd.ARPA)
```

"Stupidity, like virtue, is its own reward"

Search RISKS using swish-e

Report problems with the web pages to the maintainer

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 39

Tuesday, 19 August 1986

Contents

- Nuclear false alarm
  Robert Stroud
- Risk to beer production?
  Robert Stroud
- Re: High Tech Sex
  Lindsay F. Marshall
- QA on nuclear power plants and the shuttle
  Roy Smith
- Hackers in BITNET
  Sterling Bjorndas
- Info on RISKS (comp.risks)

Nuclear false alarm

Robert Stroud <robert%kelpie.newcastle.ac.uk@Cs.Ucl.AC.UK>
Mon, 18 Aug 86 17:56:25 bst

"BT is blamed for HM's displeasure" (Computing, August 14th 1986)
by Angus McCrone (c) Computing

British Telecom (BT) is being blamed for a network fault which caused nuclear attack sirens in Edinburgh to blare into action last month. The sirens disturbed thousands of people at 7.30 in the morning. The incident coincided with a visit by the Queen and Margaret Thatcher to watch the Commonwealth Games.

A spokeswoman at the Home Office, which has the responsibility for civil defence in the UK, said that BT was checking a carrier control unit in Edinburgh. This is believed to have malfunctioned causing the alarm to go off. The carrier control unit, one of about 250 around the country, has the job of connecting the Ministry of Defence's air operations computer centre and local police stations which activate the alarm.

The Home Office has ruled out computer error as a reason for the mistake, and seems convinced that human error or sabotage were not involved either. This is despite the fact that no similar mistakes have been recorded in the past 12 years, and that the incident happened at the height of a controversial visit to Scotland by the Prime Minister. A BT official confirmed that a report on the alarm had been sent to the Home Office, but would not say whether his company accepted responsibility for the mistake.

In time of war the Home Office consults with the MoD before ordering police stations to switch on the alarms, which warn citizens to expect air or nuclear attack. The incident in Edinburgh last month caused little panic because most people switched on their radios to check there was no real emergency.

Risk to beer production?

Robert Stroud <robert%kelpie.newcastle.ac.uk@Cs.Ucl.AC.UK>
Mon, 18 Aug 86 17:52:47 bst

"Minor bug starts mass beer panic" (Datalink, August 11th 1986)
by Dave Holmes (c) Datalink

A holidaying programmer sparked off a bizarre series of events last week culminating in a rumour that Tetley's, Yorkshire's most famous brewery, had stopped production. Workmates had realised that they needed the advice of the programmer, Richard Chubb, to sort out a problem with the control system he was developing for Tetley. Police were asked to help track him down on holiday in Scotland, but a telex from Strathclyde police to seven Scottish police divisions apparently suggested that the brewery had stopped production because of a computer breakdown.

News of this got back to Yorkshire and last weekend Tetley was deluged with calls from worried publicans afraid that supplies of Yorkshire's finest were about to dry up. David Gaskill, of the engineering company Turkington which was installing the control system explained what had happened: "There was a communications glitch between two systems we are installing at Tetley, and the program is not fully documented yet. To go through the code was going to take ages, but Richard could have sorted it out in 20 minutes," he said.

Re: High Tech Sex

"Lindsay F. Marshall" <lindsay%kelpie.newcastle.ac.uk@Cs.Ucl.AC.UK>
Tue, 19 Aug 86 15:21:58 bst

The interesting question that it raises is that of what has happened to the information on the data-base. Has it been destroyed, or has it been incorporated into the Police computer records?

Lindsay

[The implication of the article was that indeed the records had been confiscated. With a shredder in the office, it could have been what was on the diskettes -- but more than likely there were simply printouts lying around. PGN]

QA on nuclear power plants and the shuttle

Roy Smith <allegra!phri!roy@seismo.CSS.GOV>
Tue, 19 Aug 86 11:50:39 edt

Last night I watched "The China Syndrome" on TV. For those of you not familiar with this moderately-trashy movie, it's about the threat of a meltdown at a nuclear power plant. It seems that when the plant was built, the X-ray testing of the welds was faked, so a bad weld went unnoticed (causing a pump to fail, etc).

[That was taken from some real cases... PGN]

Anyway, at one point, the hero exclaims, "but our quality control is second only to NASA's!" Shows you the RISKS of making comparisons, doesn't it? Do nuclear plants have O-rings?

[No, but they do have lots of reports of equipment failures and human errors that don't seem to get wide public view. PGN]

Hackers in BITNET

<BJORNDAS%CLARGRAD.BITNET@WISCVM.WISC.EDU>
18 AUG 86 12:43-PST

The following is an abridged version of an article from issue 3.3 of VM-COM, an e-magazine published distributed in BITNET. It has been edited with permission, by Sterling Bjorndahl (BJORNDAS@CLARGRAD).

Life in the Fast Lane: Column #2

Chris Condon BITLIB@YALEVM

There are hackers in BITNET. You aren't surprised, I'm sure. Now, not all hackers are slavering, demented, animals waiting to break into, crash, and destroy systems, illegally using their resources, plundering userids that are not their own, and making a general mess out of everything.

Only some are.

There exists in this network a group of hackers who broke into a userid at Fermilab via BITNET. They used the RELAY conference machine system to keep in contact. Administration types at Cornell University, hearing of this, came to this conclusion:

"The Cornell Relay has been shut down forever due to the misuse of BITNET by some hackers in West Germany who discussed their trade on the Relay. It is Cornell's desire to not be associated with the Relay system in the future..."

The reaction by these people might seem a bit extreme, but it could be even worse. There are some people in BITNET who would like to see students completely banned from the network, or chatting banned from the network, or both. These are people to be reckoned with. They are in positions of power to do such things at their own nodes, given enough reason. For Cornell, the hackers breaking into Fermilab turned out to be an excellent excuse. It need not be anything so extreme.

Our actions are a reflection on the students in BITNET. It has been said (not enough) that BITNET usage is a privilege. It brings with it a great responsibility. Everything we do may have far reaching effects without our knowing it. The hackers that broke into Fermilab were not from Cornell, had no intention of getting that Relay shut down, and they probably did not consider that it would happen.

I posted a notice on this subject for the Usage Guidelines Group via LISTSERV@BITNIC. These are some of the responses (names withheld):

A. "The problem, as I see it, stems from a lack of moral and ethical standards in the computer world, as well as the natural inquisitiveness of young people specifically and computer type people generally."

[I disagree that "computer types" have any worse ethical standards than the bulk of this society. They just have a lot of power. - S.B.]

B. "I don't know what, if any, audit trail is left from interactive traffic on the net. If there isn't any, I think there ought to be and installations with security concerns about chatting should monitor the traffic for suspicious activity."

C. "A totally restrictive policy, one that makes absolute and unbending restrictions, especially to undergraduate students, will have two effects.

1: Those persons who are borderline on being responsible or abusive with the system may just go the wrong way, partly out of challenged to their perception of a "cold-hearted" system.

2: Students will lack (unless they break in and get away with it which is what we try to prevent) a practical education of how real life computers are implemented. I know these things to be true from first hand experience, because I used to be such a hacker. I did get away with it and I did learn enough to go right into an upper level systems programming job right out of school... The school I attended had a very closed policy. They were, however, not effective in implementing that policy, and so some of us got into the system."

D. "My suggestion is that a policy be established to deal [constructively] with "curious students" who show promise. Just how you do this has to depend on your resources."

Like it or not, someone is looking over your shoulder. Maybe you won't get caught when you do something irresponsible via BITNET, but somebody will pay the consequences. Somebody out there is looking for an excuse to shut you, or some other student, out of BITNET... The actions of some students have simply led him to believe that shutting students out is a good thing. It will take your example to convince him otherwise.

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 40

Thursday, 21 August 1986

Contents

- QA on nuclear power plants and the shuttle
  - Eugene Miya
  - Ken Dymond
- CAD, Simulation, Armored Combat Earthmover, and Stinger
  - Mary C. Akers
- Risks Distribution List -- Private-Copy Subscribers PLEASE READ!
  - PGN
- Could computers launch a nuclear attack?
  - Jeff Myers
- Info on RISKS (comp.risks)

Re: QA on nuclear power plants and the shuttle (re "portary"-als)

Eugene Miya <eugene@AMES-NAS.ARPA>
20 Aug 1986 1045-PDT (Wednesday)

> Date: Tue, 19 Aug 86 11:50:39 edt
> From: allegra!phri!roy@seismo.CSS.GOV (Roy Smith)
>
> ... I watched "The China Syndrome" on TV... moderately-trashy movie...
> Anyway at one point, the hero exclaims, "but our quality control
> is second only to NASA's!" Shows you the RISKS of making comparisons,
> doesn't it? Do nuclear plants have O-rings?
>
>   [No, but they do have lots of reports of equipment failures
>   and human errors that don't seem to get wide public view. PGN]

Risks of films?

I saw China Syndrome the day TMI occurred. It is a reasonably accurate film, with a minimum of dramatic license (the "vibration" is an example of this as control rooms tend to be more isolated.). I don't regard the film as trashy. There are deliberate attempts by film makers to be "realistic", and this film was well researched. In contrast, War Games looked trashy to computer people. The screenplay writers gave a talk about the film at the Palo Alto CPSR meeting. They deliberately used obsolete hardware so that companies like A*e might not sue them.

Sorry, Peter, you are wrong. Reactors do use O-rings. Your car uses O-rings; one just failed in my VW Rabbit. The problem of reporting is historical and dates back to the late 40s and the "mysticism" on about nuclear information. It is very easy to classify nuclear information: for instance, it is not forbidden to have civilians in any nuclear control room (they are not much different from coal fired plants in layout). This was driven by the concern for nuclear terrorism in the late 1970s. It boils down to whether nuclear power should be under civilian or military control: I know civilian physicists at LLNL who think the original decision in the 1940s was a mistake. (They feel it should have been kept a military secret.)

NASA's QA. I've not worked on QA. The problem might be in the Q: The paperwork for individual Shuttle tiles weigh more than the tiles themselves. There is a photo in Scott Crossfield's autobiography (1964?) showing paperwork for the X-15 exceeding 3 times the weight of the X-15. We must not mistake quantity for real quality. Maybe software should have more paper.... Let's not confuse quantitative assurance and qualitative.

Lastly, (here's the nerve you hit), Hans Mark (currently head of the U of Texas) gave a talk at Ames on Monday on Challenger and Chernobyl. Hans is and was in a unique position to talk about both. He was a chief at LLL, taught nuclear engineering at UCB for 10 years, ran Ames, ran the Air Force, #2 man at NASA and made flight decisions for the first dozen flights (O-ring charring on fights 2, 8 and later). He was interviewed by the Rogers Commission. "O-rings, did not seem like that much of a problem in contrast to other problems like nozzle burn thru..." Mark has decided to write an article based on this talk.

He feels somewhat responsible even though he is no longer with NASA. He had scheduled a review regarding O-rings during a period when he took his new U-Texas job. The review never took place. (Lame duck administrator, in his words.) The men who made the final launch decisions were and still are friends of his. The Chernobyl portion was a recapping of known information. In both cases, Mark cites the need for communication between management and workers.

--eugene miya
NASA Ames Research Center
eugene@ames-aurora.ARPA

[I saw it the NIGHT BEFORE TMI! But I asked Gene about whether those other O-rings also had problems at low temperatures. (PGN) This was Gene's reply:]

Cars: Mine was 8 years old. It was an external seal, it failed at 80 degs F.

Power plants: probably not.

I would think antarctic snow cars have O-rings and fan belts and all sorts of things that snap.

--eugene

Re: QA at Nuclear Plants

"DYMOND, KEN" <dymond@nbs-vms.ARPA>
21 Aug 86 09:41:00 EDT

PGN comments in [RISKS 3-39](http://catless.ncl.ac.uk/Risks/3.39.html) on "QA on nuclear power plants and the shuttle":

> >No, but they [nuclear power plants] do have lots of reports
> >of equipment failures and human errors that don't seem to
> >get wide public view.

It may depend on how interested the public is. These reports (and probably PGN is referring to the Licensee Event Reports or LERs which are compiled by the NRC from plants, i.e. holders of licenses to make electricity from nuclear power) are matters of public record. The NRC distributes them to all plants as notices of the kinds of things that happen and should be watched for. They are also maintained in the NRC's public documents room in the D.C. area and in a local public documents room near every nuclear plant. I know of at least one public library (Wiscasset, Maine) that keeps LERs on file because of public interest in the Maine Yankee plant nearby.

Most of the time LERs don't make exciting reading. I haven't seen an LER for a while but a representative incident that comes to mind occurred at a plant where the fuel tanks for the emergency diesel generators were allowed to get 300 gallons low (out of 3000 or 30000 gals., can't remember). Some fuel is used up in the weekly test of making sure the generators start and operate and I guess the tanks are supposed to be topped up. The 10 percent or so shortfall of fuel would have been remedied at the next (I think it was weekly) scheduled visit from the oilman. I don't remember whether the NRC levied a fine in this case.

The LERs serve as a record of errors in the industry, something that would be a great help if it existed for software engineering. Civil and structural engineers investigate structural failures and publish detailed results of the investigations in their literature, another practice that software engineers might consider.

The LERs are supposed to be exhaustive and one thing the resident NRC inspector at every plant does is to make sure that all events required by regulations to be reported do get reported. If the story about the defective welds is true, it should be in an LER somewhere.

Ken Dymond

CAD, Simulation, Armored Combat Earthmover, and Stinger

"Mary C. Akers" <makers@cct.bbn.com>
Thu, 21 Aug 86 10:26:23 EDT

Recently the Risks list had a short discussion on the excessive use of CAD systems. The September 1986 issue of Discover Magazine has an article by Wayne Biddle on the use and abuse of computer modeling and simulation. It is entitled "How Much Bang for the Buck?" Here are a few interesting quotes:

"I want to replace sterile computers simulations with more realistic testing under combat conditions," says Representative Charles Bennett of Florida, [...]"Weapons testing should ensure that our weapons work in combat, not just in the laboratory." With that statement, Bennett zeroes in on the main bone of contention among those concerned with weapons design and testing: whether computer simulation and modeling can take the place of live trails with real equipment."

"The thing we worry about most is validating our simulations (that is, proving they're realistic), and validation is lagging, for sure. Without test data, an unvalidated simulation is all we have."

"Simulated Flying is so different from real flying that the Navy finds that working in a simulator can be a detriment to safe operation of an airplane."

Some of the examples used in the article include:

The Army's Armored Combat Earthmover (ACE) - "...which underwent 18,000 hours of testing without ever being operated under field conditions. [When it finally under went live trails at Fort Hood] ...the tests revealed that the ACE's transmission cracked, that is muffler caught fire, that the driver's hatch lid was too heavy to lift, and that doing certain maintenance work "could endanger the operator's life."

"The Stinger, a 'man-portable' ground-to-air missile, proved too heavy for soldiers to carry on long marches; gunners must hold their breath after firing to avoid noxious fumes."

Risks Distribution List -- Private-Copy Subscribers PLEASE READ!

Peter G. Neumann <Neumann@CSL.SRI.COM>
Wed 20 Aug 86 11:04:45-PDT

One of our readers asked to be removed from the RISKS list, forwarding this somewhat heavy-handed note from an administrator at his institution:

"Please unsubscribe from the lists you have joined. At [...] individuals do not join mailing lists directly. There will be a way for you to read the full distribution of lists in the fall. For now I must ask you to stop receiving your own copies of everything."

When RISKS began a year ago, the initial intent was to provide individual subscriptions only until appropriate BBOARDs could be set up. For the convenience of some individuals, we have continued to provide private copies. The local mailer overhead attributable to RISKS is nontrivial -- although the new intelligent mailers cut down on net traffic. Disk storage is now approaching 800 DEC-20 pages for the full collection to date. Maintenance of the RISKS list continues to be a problem with all the address changes, incessant notifications of individual nondeliveries (sorry if we overflow your disk quotas!), host outages, etc. [Welcome back, Dockmaster -- which took months to recover from lightning hitting their IMP.] Unfortunately, various BBOARDs have allocated enough space for only a few recent back issues (presumably on the assumption that the earlier issues can be FTPed or that they lose their timeliness).

If you receive a private copy and could conveniently be reading RISKS on a local BBOARD, please ask me to remove you from the list. Thanks... Peter

---

Could computers launch a nuclear attack?

Jeff Myers <myers@unix.macc.wisc.edu>
Thu, 21 Aug 86 09:41:49 cdt

> [NEW ARTICLE ON OLD TOPIC. Earlier followers of this story may wish to read the last three paragraphs. PGN]

[from the August 20 *Guardian*, p. 9]
By Dave Kadlecek, *Guardian* Bureau

SAN FRANCISCO -- A Stanford University computer professional has sued Secretary of Defense Caspar Weinberger, claiming that government plans allowing computers to automatically launch a nuclear attack are unconstitutional.

Clifford Johnson, a manager in Stanford's Information Technology Services, filed the suit in federal district court in San Francisco June 17. He charged that the US government has a policy of operating a launch-on-warning capability, under which the US would launch a retaliatory nuclear attack against the USSR on the basis of a warning that Soviet missiles are on the way, before unequivocal confirmation that an attack actually occurred. Due to the short times involved, such a launch capability relies upon computerized warning systems which are prone to error and cannot allow for meaningful human intervention in a launch decision.

This automatic decision illegally usurps congressional powers and delegates presidential powers. Thus, Johnson's suit argues, the resulting ``likelihood of a nuclear counterstrike and global environmental damage'' would deprive Johnson of life and property without due process of law, giving him standing to sue now, since it would not be possible to do so after a nuclear war. He asked that the court declare that the secretary of defense's oath of office ``obligates him to forthwith cease and desist from operating his launch-on-warning capability.''

Under a cautious assumption that launch-on-warning is in continuous use only during crisis situations, a number of studies have predicted that an accidental nuclear war is statistically likely within the next 30 years.

Johnson maintains, however, that US policy already does continuously use launch-on-warning capability by any normal interpretation of the word ``policy,'' but this denial means only that a formal decision will not be made until a button is pushed when the warning occurs. Indeed, a highly sophisticated set of procedures and programs for a launch-on-warning is in continuous operation, guarding against a feared ``bolt-from-the-blue'' attack by short-range submarine-launched ballistic missiles. The Single Integrated Operational Plan consists of a menu of nuclear ``attack options'' -- lists of targets with assignments of weapons to hit them. The plan contains launch-on-warning options, and procedures now in operation permit the selection of a launch-on-warning option in response to a surprise attack.

In support of Johnson's suit, Computer Professionals for Social Responsibility (CPSR) emphasize the inevitability of some computer error in a system as complex as a launch-on-warning system. The most dangerous computer errors are not failures of the device itself (hardware errors), but of the programming (software errors), stemming ``not from inadequacies in the technology, but rather from the inability of human beings to formulate totally adequate plans (programs) for dealing with complicated, poorly understood situations,'' says CPSR. CPSR is ``concerned that the government is pursuing a launch-on-warning capability, in the mistaken belief that computer technology can safely be entrusted with important decisions regarding the release of nuclear weapons. If this course is allowed to continue unchecked, it is only a matter of time before a catastrophic error occurs.''

GROUPS IN SUPPORT

Though not an attorney, Johnson filed suit on his own behalf, and will argue his own case through the resolution of government motions to dismiss the suit, on which hearings are expected this fall. However, he will need to hire a lawyer if the case goes to trial, and the Lawyer's Alliance for Nuclear Arms Control (LANAC) and the Center for Constitutional Rights have agreed to help at the appellate level.

In addition to CPSR, support has come from peace groups and from former aerospace engineer Robert Aldridge, coauthor of ``First Strike'' and co-editor of ``The Nuclear Time Bomb,'' and constitutional scholar Arthur Miller.

Johnson had filed a similar suit in 1984. He lost in district court when the judge ruled that it was a political matter, not for the judiciary to decide. His appeal was rejected, not by upholding the lower court's reasoning, but by ruling that since he then claimed only that the government had a launch-on-warning capability, not necessarily a launch-on-warning policy, the unused capability was not a threat over which he could sue.

Johnson's current suit includes sensitive information he had deliberately excluded from his earlier suit, such as evidence that the Strategic Air

Command possesses the authorization codes needed to launch a nuclear attack.

``I've gone back, I've done my homework, I say we've got launch-on-warning now and I'm prepared to prove it,'' said Johnson. ``We're at peace, so why risk my neck?''

Search RISKS using swish-e

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 41

Saturday, 23 August 1986

Contents

- $1 million bogus bank deposit
  Hal Perkins
- Cheating of automatic teller machines
  Jacob Palme
- Simulation, Armored Combat Earthmover, and Stinger
  Herb Lin
- Report from AAAI-86
  Alan Wexelblat
- Info on RISKS (comp.risks)

$1 million bogus bank deposit

Hal Perkins <hal@gvax.cs.cornell.edu>
Fri, 22 Aug 86 21:47:58 EDT

From the Chicago Tribune, Friday, Aug. 15, 1986. sec. 3, p. 3:

Bank machine is no match for schoolboy with a lollipop

AUCKLAND, New Zealand [UPI] -- A schoolboy outsmarted an automatic bank machine by using the cardboard from a lollipop packet to transfer $1 million New Zealand dollars into his account, bank spokesmen said Thursday.

Tony Kunowski, corporate affairs manager of the United Building Society savings and loans institution, said the 14-year-old student slipped the cardboard into an envelope and inserted it into the machine while punching in a deposit of $1 million, the U.S. equivalent of $650,000.

"We are not amused, but we don't think this is the tip of an iceberg," he said of the incident of three weeks ago.

Kunowski said that when the boy, identified only as Simon, checked his account a few days later, he was amazed to discover the money had been credited. He withdrew $10.

When no alarm bells rang and no police appeared, he withdrew another $500. But his nerve failed and he redeposited the money.

On Tuesday, Simon withdrew $1,500, Kunowski said.

But his nerve failed again Wednesday, and he told one of his teachers at Selwyn College, Kunowski said. The school's headmaster, Bob Ford, took Simon to talk with United Building Society executives.

Ford said Simon had not been considered one of his brightest pupils, "at least until now."

It was unknown if Simon would be disciplined.

Kunowski told reporters that Simon succeeded because of delays in reconciling transactions in automatic tellers around the country with United's central computer system.

"The delay in toting up the figures would normally be four weeks and that was how a schoolboy could keep a fake million dollars in his account without anyone batting an eyelid," he said.

"We are now looking very closely at our internal systems. Human error may also be involved," Kunowski said.

Cheating of automatic teller machines

<Jacob_Palme_QZ%QZCOM.MAILNET@MIT-MULTICS.ARPA>
21 Aug 86 02:45 +0200

Several young people have cheated automatic teller machines from one of the largest Swedish bank chains in a rather funny way.

You use the machines by inserting your plastic card in a slot, then punching the amount you want and your password, and then the card comes out of one slot, and the money out of another slot.

The cheaters took a badge belonging to a large guard company, which looked very reassuring, and fastened it with double-sticky tape in front of the slot through which money comes out. They then faded into the background and waited until someone came to get money from the machine. The person who wanted to use the machine put in his card, punched his code and amount, and the machine started to push out the money through the slot. When the money could not get out, because of the obstruction, the machine noted this, and gave a "technical error" message to the customer, who went away. Up came the youngsters, who took away the badge, fetched the money behind it, and put up the badge again for the next customer.

The cheatings described above have been going on for several months, but the bank has tried to keep this secret, claiming that if more people knew about, more would try to cheat them. Since the money is debited on the account of the customers, this means that those customers who did not complain lost the money. The bank has now been criticised for keeping this secret, and has been forced to promise that they will find all customers cheated (this is possible because the temporary failure in getting the money out of the slot was noted automatically by the machine) and refund the money lost.

The bank chain will now have to rebuild 700 automatic dispensing machines. Most other banks in Sweden, except this chain, have a joint company operating another kind of dispensing machines, from which you can take out money from your account in any of these banks. Their dispensing machines cannot be cheated in this way, because they have a steel door in front of the machine which does not open until you insert a valid plastic card.

Simulation, Armored Combat Earthmover, and Stinger

<LIN@XX.LCS.MIT.EDU>
Fri, 22 Aug 1986 08:53 EDT

From: Mary C. Akers

Report from AAAI-86 [Really from Alan Wexelblat]

Fri, 22 Aug 86 13:05:57 CDT

I just got back from a week at AAAI-86. One thing that might interest RISKS readers was the booth run by Computer Professionals for Social Responsibility (CPSR). They were engaged in a valiant (but ineffectual) effort to get the AI mad-scientist types to realize what some of their systems are going to be doing (guiding tanks, cruise missiles, etc.).

They were handing out some interesting stuff, including stickers that said (superimposed over a mushroom cloud): "It's 11 p.m. Do you know what your expert system just inferred?"

They also had a series of question-answer cards titled "It's Not Trivial." Some of them deal with things that have come up in RISKS before. [I left them in for the sake of our newer readers. PGN] They are:

Q1: How often do attempts to remove program errors in fact introduce one or more additional errors?

A1: The probability of such an occurance varies, but estimates range from 15 to 50 percent (E.N. Adams, "Optimizing Preventing Service of Software Products," _IBM Journal of Research and Development_, Volume 28(1), January 1984, page 8)

Q2: True or False: Experience with large control programs (100,000 < x < 2,000,000 lines) suggests that the chance of introducing a severe error during the correction of original errors is large enough that only a small fraction of the original errors should be corrected.

A2: True. (Adams, page 12)

Q3: What percentage of federal support for academic Computer Science research is funded through the Department of Defense?

A3: About 60% in 1984. (Clark Thompson, "Federal Support of Academic Research in Computer Science," Computer Science Division, University of California, Berkeley, 1984)

Q4: What fraction of the U.S. science budget is devoted to defense-related R&D in the Reagan 1985/86 budget?

A4: 72% ("Science and the Citizen," _Scientific American_ 252:6 (June 1985), page 64)

Q5: The Space Shuttle Ground Processing System, with over 1/2 million lines of code, is one of the largest real-time systems ever developed. The stable release version underwent 2177 hours of simulation testing and the 280 hours of actual use during the third shuttle mission. How many critical, major, and minor errors were found during testing? During the mission?

A5:

	Critical	Major	Minor
Testing	3	76	128
Mission	1	3	20

(Misra, "Software Reliability Analysis," _IBM Sys. J. 1983, 22(3))

Q6: How large would "Star Wars" software be?

A6: 6 to 10 million lines of code, or 12 to 20 times the size of the Space Shuttle Ground Processing System. (Fletcher Report, Part 5, page 45)

The World Wide Military Command and Control System (WWMCCS) is used by civilian and military authorities to communicate with U.S. military forces in the field.

Q7: In November 1978, a power failure interrupted communications between WWMCCS computers in Washington, D.C. and Florida. When power was restored, the Washington computer was unable to reconnect to the Florida computer. Why?

A7: No one had anticipated a need for the same computer (ie the one in Washington) to sign on twice. Human operators had to find a way to bypass normal operating procedures before being able to restore communications. (William Broad, "Computers and the U.S. Military Don't Mix," _Science_ Volume 207, 14 March 1980, page 1183)

Q8: During a 1977 exercise in which WWMCCS was connected to the command and control systems of several regional American commands, what was the average success rate in message transmission?

A8: 38% (Broad, page 1184)

Q9: How much will the average American household spend in taxes on the
military alone in the coming year?

A9: $3,400 (Guide to the Military Budget, SANE)

[question 10 is unrelated to RISKS]

Q11: True or False? Computer programs prepared independently from the same
specification will fail independently.

A11: False. In one experiment, 27 independently-prepared versions, each
with reliability of more than 99%, were subjected to one million
test cases. There were over 500 instances of two versions failing
on the same test case. There were two test cases in which 8 of the
27 versions failed. (Knight, Leveson and StJean, "A Large-Scale
Experiment in N-Version Programming," Fault-Tolerant Computing
Systems Conference 15)

Q12: How, in a quintuply-redundant computer system, did a software error
cause the first Space Shuttle mission to be delayed 24 hours only
minutes before launch?

A12: The error affected the synchronization initialization among the 5
computers. It was a 1-in-67 probability involving a queue that
wasn't empty when it should have been and the modeling of past
and future time. (J.R. Garman, "The Bug Heard 'Round the World,"
Software Engineering Notes Volume 6 #5, October 1981, pages 3-10)

Q13: How did a programming punctuation error lead to the loss of a Mariner
probe to Venus?

A13: In a FORTRAN program, DO 3 I = 1,3 was mistyped as DO 3 I = 1.3 which
was accepted by the compiler as assigning 1.3 to the variable DO3I.
(_Annals of the History of Computing_, 1984, 6(1), page 6)

Q14: Why did the splashdown of the Gemini V orbiter miss its landing point
by 100 miles?

A14: Because its guidance program ignored the motion of the earth around
the sun. (Joseph Fox, _Software and its Development_, Prentice Hall,
1982, pages 187-188)

[Questions 15-17 are not RISKS related]

Q18: True or False? The rising of the moon was once interpreted by the
Ballistic Missile Early Warning System as a missile attack on the US.

A18: True, in 1960. (J.C. Licklider, "Underestimates and Overexpectations,"
in _ABM: An Evaluation of the Decision to Deploy and Anti-Ballistic
Missile_, Abram Chayes and Jerome Wiesner (eds), Harper and Row,
1969, pages 122-123)

[question 19 is about the 1980 Arpanet collapse, which RISKS has discussed]

Q20: How did the Vancouver Stock Exchange index gain 574.081 points while the stock prices were unchanged?

A20: The stock index was calculated to four decimal places, but truncated (not rounded) to three. It was recomputed with each trade, some 3000 each day. The result was a loss of an index point a day, or 20 points a month. On Friday, November 25, 1983, the index stood at 524.811. After incorporating three weeks of work for consultants from Toronto and California computing the proper corrections for 22 months of compounded error, the index began Monday morning at 1098.892, up 574.081. (Toronto Star, 29 November 1983)

Q21: How did a programming error cause the calculated ability of five nuclear reactors to withstand earthquakes to be overestimated, and the plants to be shut down temporarily?

A21: A program used in their design used an arithmetic sum of variables when it should have used the sum of their absolute values. (Evars Witt, "The Little Computer and the Big Problem," AP Newswire, 16 March 1979. See also Peter Neumann, "An Editorial on Software Correctness and the Social Process," _Software Engineering Notes_, Volume 4(2), April 1979, page 3)

Q22: The U.S. spy ship Liberty was attacked in Israeli waters on June 8, 1967. Why was it there in spite of repeated orders from the U.S. Navy to withdraw?

A22: In what a Congressional committee later called "one of the most incredible failures of communications on the history of the Department of Defense," none of the three warnings sent by three different communications media ever reached the Liberty. (James Bamford, _The Puzzle Palace_, Penguin Books, 1983, page 283)

Q23: AEGIS is a battle management system designed to track hundreds of airborne objects in a 300 km radius and allocate weapons sufficient to destroy about 20 targets within the range of its defensive missiles. In its first operational test in April 1983, it was presented with a threat much smaller than its design limit: there were never more than three targets presented simultaneously. What were the results?

A23: AEGIS failed to shoot down six out of seventeen targets due to system failures later associated with faulty software. (Admiral James Watkins, Chief of Naval Operations and Vice Admiral Robert Walters, Deputy Chief of Naval Operations. Department of Defense Authorization for Appropriations for FY 1985. Hearings before the Senate Committee on Armed Services, pages 4337 and 4379.)

Well, this message is long enough; I'll hold off on my personal commentaries. People wanting more information can either check this sources given or contact CPSR at P.O. Box 717, Palo Alto, CA 94301.

--Alan Wexelblat

ARPA: WEX@MCC.ARPA or WEX@MCC.COM
UUCP: {ihnp4, seismo, harvard, gatech, pyramid}!ut-sally!im4u!milano!wex

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 42

Monday, 25 August 1986

Contents

- Re: $1 million bogus bank deposit
  Barry Shein
- Sometimes things go right
  Matt Bishop
- Re: Cheating of automatic teller machines
  Dave Farber
- Keystroke Analysis for Authentication
  rclex
- Computer Vote Counting In the News -- More
  John Woods
- Info on RISKS (comp.risks)

Re: $1 million bogus bank deposit

Barry Shein <bzs@BU-CS.BU.EDU>
Sat, 23 Aug 86 20:14:53 EDT

> "We are now looking very closely at our internal systems. Human
>error may also be involved," Kunowski said.

There's that term "human error" again. Note Chernobyl, TMI, etc. They also seemed to like to speak of "human error".

Is this a new form of excuse? Is it supposed to have PR value? What else? Alien-life-form error? Supernatural error?

I know most of you agree with me, and this is essentially trite. I am just starting to sensitize badly to this techno-speak.

-Barry Shein, Boston University

[I have commented on this on various occasions. Many of the problems
that we find are deeper sorts of "human error" -- the requirements
are established badly (the DIVAD?), the design is flawed (Challenger

booster rockets), the implementation is faulty (the first Shuttle launch), the patch was put in wrong (Viking), the system permits operation in an unsafe mode (Sheffield), etc. Those are clearly human errors, but they get treated in the opposite way -- not treated as human errors, but rather disanthropomorphized as "computer errors"! What you are saying is both essentially trite and very deep, both at the same time. PGN]

Sometimes things go right

Matt Bishop <mab@riacs.ARPA>
Mon, 25 Aug 86 08:19:14 -0700

All these letters about ATM's being outsmarted reminds me of an incident where someone gambled on the inability of a bank to change the programming for managing ATM's, and lost. This incident is described in Donn Parker's book on computer crime, which I seem to have left at home (so I can't give a reference), and it's interesting because it shows the risks in assuming things can't be done quickly.

In Japan, someone kidnapped a little girl, and told her father to open an account at a bank which had ATM's throughout Tokyo, and put the ransom in that account. He was then to indicate the account number and password (in the newspaper via what Sherlock Holmes would call the agony column, I guess). The kidnapper would then withdraw the money from one of the ATMs. He figured there weren't enough police to watch all the ATMs and even if there were, they would have no way of distinguishing him from any of the other patrons who made legitimate withdrawals.

Unfortunately for him, when the bank heard about this, they got several programmers together and working all night they changed the program controlling the ATMs to trap any transactions for that particular account, and immediately notify the operators at which ATM the withdrawal was taking place. They then put police at as many ATMs as they could. The father made the deposit, the kidnapper withdrew the money, and before he could get out of the ATM booth the police grabbed him. The girl was recovered safely. The programmers got a medal. The kidnapper went to jail.

Kind of nice to know that sometimes things do go wrong for the better!

Matt Bishop

Re: Cheating of automatic teller machines

Dave Farber <farber@huey.udel.EDU>
Sat, 23 Aug 86 17:01:38 -0400

That's the modern analog to the favorite telephone trick, stuff cotton [or chewing gum] up the coin return, and come back latter to collect the coin returns. (It's harder to do with the new pay phones, but not impossible.)

[Yes, many of the current tricks are reincarnations of earlier ones.
But, as we get higher-tech, new tricks are emerging as well. PGN]

Keystroke Analysis for Authentication (Re: RISKS-3.31)

<hplabs!caip!harvard!rclex!cdx39!jc@ucbvax.Berkeley.EDU>
Wed, 20 Aug 86 10:07:37 edt

> ... One gray area is checking
> the match between credentials and credential-holders: this generally has
> to be done by humans unless the credentials are something like retinagrams.

Actually, this is easier to automate than most people would guess.

A few years back, I saw a demo of one solution, which is as accurate as retinagrams, but is non-invasive. This was the measurement of a "typing profile" as a person typed something (it didn't much matter what) on a keyboard that recorded and reported microsecond-precision timing info on keystrokes.

The idea was to make a list of the most common 2-character pairs (th, he, st, se, ...), calculate ratios of the top entries (th/he, he/st, th/st, ...), and normalize by dividing throughout by the mean value of the most common pairs. The resulting histogram turns out to be quite as specific as retinagrams and fingerprints, and even harder to counterfeit.

Since then, I've been watching for applications, and have found instead that most people 1) have never heard of it, and 2) don't believe that it works. The people doing the demo weren't very concerned about either of these "problems". After all, only the ones making the decision to install it need know about it; it's better if the subject not know or understand the security system. As for the second point, it doesn't really matter whether the subject believe in it; it works regardless.

It's surprising how short a message it works with. Obviously, you need at least 3 characters; it turns out that you don't need more than about 10. Of course, there are failures. But from a security viewpoint, they are in the right direction of labeling a person as "unknown", typically when they are typing irregularly due to fatigue or drugs.

The demo system had no sign-on. You just started typing commands; the machine determined for each command who had typed it and whether the person was authorized to do what was asked. In particular, they liked to show an operator's console sitting in a non-secure area. The machine would obey commands typed by authorized operators, but not by anyone else. It was rather cute. A lot of people who tried using it got very nervous looks on their faces. "The machine really does know who I am, doesn't it?"

Of course, you couldn't use this approach with just any commercial terminal. How could you get the timing figures out of a VT100, for example? But the data collection is well within the capabilities of the typical intelligent terminal with an 8-bit micro as a controller.

I've occasionally wondered whether there are any other non-invasive identification techniques that are anywhere nearly as effective as this one. I haven't heard of any. But then, they might not be very widely advertised if they do exist.

I've also wondered about the feasibility of using this a a "user friendliness" feature. Imagine not needing to sign on to a system; you just walk up to any terminal and start typing commands....

Computer Vote Counting In the News [SOME NEW STUFF, SOME OLD]

John Woods <jfw@EDDIE.MIT.EDU>
Sat, 23 Aug 86 21:13:24 EDT

[SEE SUMMARY OF EVA WASKELL'S EARLIER TALK BY RON NEWMAN in RISKS-2.42]

Use of computers in elections raises security questions
Boston Globe, 23 August 1986, page 17
By Gregory Witcher, Globe Staff

The computer programs that will be used to count the votes in elections this fall accross the United States, including a quarter of the votes in Massachusetts, are vulnerable to tampering and fraud, according to computer specialists, researchers, science writers and attorneys.

Although no case of computer fraud has been proved, specialists say a large potential exists because of the lack of mandatory federal or state security guidelines to prevent it.

In addition, they say, there are no independent means of auditing programs to verify they are working properly and most local election officials lack the computer skills necessary to detect if computer programs are secretly altered.

"It's like a black box," says Eva Waskell, a Reston, Va., science writer who helped organize a recent two-day conference at Boston University on the potential of computer fraud in voting. "Election officials have no hard data to back their claims that these vote-counting programs are counting accurately."

Sixty-five percent of the votes cast by Americans in the 1984 presidential election were tabulated by computer systems, according to the Federal Election Commission. In next month's Massachusetts primary, computer programs will be used to tally the votes in 26 percent of the state's 351 election precincts, the Secretary of State's office says.

Four of every five of those votes will be tallied by a vote-counting program that has been challenged in cases now pending in state and federal courts in Indiana, West Virginia, and Maryland. In Indiana and West Virginia, the company was accused of helping to rig elections.

The program was developed by Business Records Corp., formerly Computer Election Systems, a Berkeley, Calif., company that federal election officials estimate produces more than half the computer voting equipment used nationwide. Company officials in Berkeley and Chicago could not be reached for comment yesterday.

John Cloonan, director of the elections division of the Massachusetts Secretary of State's office, said there have been no instances of computer fraud reported since Massachusetts first began using a computer-assisted voting system in 1967.

Computerized voting is now used in Massachusetts jurisdictions ranging in size from Worcester, the state's second largest city with about 80,000 registered voters, to Avon, where there are 3,000 registered voters, Cloonan said.

Voters in Boston and in one-third of all Massachusetts communities cast their ballots on mechanical lever-type machines. The remaining cities and towns use paper ballots.

According to David Stutsman, who participated in the two-day seminar at BU, a recount of the votes cast in Elkhart County, Ind., in November 1982 showed that the computer program had improperly printed the results of one race in another, failed to count all the votes for one candidate and counted 250 more votes than there were voters in a third race.

Stutsman is an attorney representing eight candidates who challenged the election results in lawsuits alleging that the vote counting was "false and fraudulent."

Stutsman contended that a computer programmer from the company changed the computer program's instructions on election night, but without a system to record changes made in the pgram and without election officials knowledgable about how the program worked, "it was impossible to say how the votes were counted and whether they were counted accurately or not."

In another case presented at the conference, a review of 1984 election results showed that President Reagan received 159 votes in the Trinity River Bottom precinct, defeating challenger Walter Mondale by a 3 to 1 margin in the Texas district inhabited only by squirrels, rabbits and fish.

"The computer invented those numbers. The numbers could not have gone into the program but they came out," said Terry Elkins, a political researcher in Dallas who studied the election results. "No one lives there, so the fish must have voted."

Despite reports like these, others remain confident that computer voting is not terribly vulnerable to fraud or error. "The smoke far outweighs the fires," William Kimberling, a federal elections administrator in Washington, said. Kimberling said that none of the allegations of fraud raised in the legal challenges has been upheld in court.

Words, words, words...

<LIN@XX.LCS.MIT.EDU>
Mon, 25 Aug 1986 15:08 EDT

> The point is that a person who believes something, however erroneously, and espouses and publicly supports that belief, is *not* lying. These are complex times. There are many matters about which reasonable persons, even reasonable scientists, may differ. There is no point in saying that a person lied when that person was doing the best work possible based on the knowledge and belief available at the time.

I'd like to believe this, but I think you leave out a major category

-- how are we to classify what could be called "deliberate ignorance"? That is probably the most charitable label that one could give to the call for SDI -- a system that will eliminate the threat of nuclear ballistic missiles. Some people (some of them on RISKS) have called such statements merely "political rhetoric". But when the call is for defense of the entire population, and NO ONE in the scientific community believes that it is possible to frustrate a deliberate Soviet attack on the U.S. population, isn't that either lying (at worst) or deliberate dumbness at best?

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 43

Tuesday, 26 August 1986

Contents

- Comment on PGN's comment on human error
  Nancy Leveson
- Keystroke Analysis for Authentication
  Scott E. Preece
  Eugene Miya
- Risks of Mechanical Engineering [More on O-Rings]
  Martin Harriman
- Re: Words, words, words...
  Mike McLaughlin
- Comments on paper desired
  Herb Lin
- Info on RISKS (comp.risks)

Comment on PGN's comment on human error

Nancy Leveson <nancy@ICSD.UCI.EDU>
26 Aug 86 11:52:32 PDT (Tue)

Both "human error" and "computer error" are meaningless words. At least in scientific discussions, we should attempt to use words that can be defined. There are not "deeper" human errors (or "shallower" ones?). There are design flaws or inadequacies, operational errors, hardware "random" (wear-out?) failures, management errors, etc. In hardware, there are also production errors. These may not be good categories, and I welcome suggestions for better ones. But if we can categorize, then it may help us to understand the issues involved in risks (by locating general themes) and to devise fixes for them.

But in doing this we must be very careful. Accident causes are almost always multifactorial. TMI, for example, involved all of the above categories of errors including several hardware failures, operator errors, management errors, and design flaws. Challenger also appears to follow the same trend. [I mention these two because they are both accidents which involved extensive investigation into the causes]. According to my friends in System Safety Engineering, this is true for ALL major accidents. As I have mentioned earlier in this forum, liability plays a major role in attempts to ascribe accidents to single causes (usually involving operator errors because they cannot be sued for billions). Also, the nature of the mass media, such as newspapers, is to simplify. This is one of the dangers of just quoting newspaper articles about computer-related incidents. When one reads accident investigation reports by government agencies, the picture is always much more complicated. Trying to simplify and ascribe accidents to one cause will ALWAYS be misleading. Worse, it leads us to think that by eliminating one cause, we have then done everything necessary to eliminate accidents (e.g. train the operators better, replace the operators by computers, etc.).

But even though it is difficult to ascribe a "cause" to a single factor, it is possible to describe the involvement of the computer in the incident, and this is what we should be doing. We also need to understand more fully the "system" nature of accidents and apply "system" approaches to preventing them. If accidents are caused by the interaction of several types of errors and failures in different parts of the system, then it seems reasonable that attempts to prevent accidents will require investigation into and understanding of these interactions along with attempts to eliminate individual problems. Elsewhere I have given examples of serious computer-related accidents that have occurred in situations where the software worked "correctly" (by all current definitions of software correctness) but where the computer software was one of the major factors ("causes") in the accident.

Since I specialize in software safety, I interact with a large number of companies and industries (aerospace, defense, medicine, nuclear power, etc.) concerned with this problem. The most successful efforts I have seen have involved companies where the software group and engineers have worked together. Unfortunately, this is rare. The majority of the people who come to my talks and classes and with whom I work are engineers. The software personnel usually argue that:

(1) safety is a system problem (not a software problem) and thus is the province of the system engineer. They are too busy doing their own work developing software to participate in system safety meetings and design reviews.

(2) they already use good software engineering practices and therefore are doing everything necessary to make the software safe.

I.e., "leave me alone and let me get back to my job of producing code, and don't waste my time by making me attend meetings with the system engineers. They can do their job, and I'll do mine."

Unfortunately, almost all of the techniques that appear to be useful in producing safer computer-controlled systems require the involvement of the software designers and implementers.

Nancy Leveson
Information and Computer Science Dept.
University of California, Irvine

Keystroke Analysis for Authentication (RISKS-3.42)

"Scott E. Preece" <preece%ccvaxa@GSWD-VMS.ARPA>
Tue, 26 Aug 86 08:59:00 cdt

I would think this would only be safe if you had physical security for the terminal -- otherwise the determined break-in artist could record the appropriate sequences and play them back as desired. Of course, if you allow that kind of intrusion any kind of password scheme is also hopeless.

scott preece, gould/csd - urbana   uucp: ihnp4!uiucdcs!ccvaxa!preece

---

Re: Keystroke Analysis for Authentication (Re: RISKS-3.31)

Eugene Miya <eugene@AMES-NAS.ARPA>
Tue, 26 Aug 86 15:02:33 pdt

We just had a demostration of the keystroke authentication system by Dr. John David Garcia. To clarify a couple of things. The shortest realistic name should be 5 characters (Ed Ng). 10 characters is better. The system uses a statistical distance function and is based on the old idea of telegraph key signatures. It is not just a matter of starting to type. A user must do between 70-80 trials to train a system to recognize a signature. A lower figure is used for touch typists. Non-typists can be recognized with a sort of relaxation phenomena when they adapt to using the system: users (believe it or not go into "an alpha state" [not my quote]) have to relax in order to consistency log in. It seems other benefits or problems result: any significant quantity of alcohol or other drug affects timing: three drinks and you can't log in [good and bad]. The mechanism for determining timing is not for general purpose typing, only particular strings. This also brings up the fact that some times you don't always log in on the first try. Garcia is speaking to various Government agencies and computer manufacturers about this system, but it would not be appropriate to say whom. Signatures tend to be keyboard specific, so trials are required for different keyboards. Despite these draw backs, the system appears quite nice. It does not require "microsecond timing," 60 Hz wall clock timing is adequate. There is probably be a demostration of the system at the next Compcon in San Francisco. The demo we saw was running on a Compauq written in BASIC with a couple of assembly language kernels.

--eugene miya;   NASA Ames Research Center;   eugene@ames-aurora.ARPA
  {hplabs,hao,dual,ihnp4,decwrl,allegra,tektronix,menlo70}!ames!aurora!eugene

---

Risks of Mechanical Engineering [More on O-Rings]

Martin Harriman <MARTIN%SRUCAD%sc.intel.com@CSNET-RELAY.ARPA>

Fri, 22 Aug 86 10:40 PDT

O-rings are used in many applications where a reliable gas or liquid seal is desired; they are generally the most reliable method for sealing a joint that must be disassembled periodically. There are lots of interesting failure mechanisms (interesting if you are a mechanical engineer), but I doubt any of them involve computers, except in the most peripheral fashion.

O-ring failures in automobiles are usually the result of hardening, either due to chemical attack (usually methanol in gasohol), or heat.

The recent failures (NASA, Chernobyl, TVA) don't have a lot to do with computers, per se--I claim each of these cases were due to poor management. In NASA's case, we have the spectacle of NASA management ignoring engineering concerns because of the pressure to launch. So NASA will listen to the WCTU (who convinced NASA to abandon their plans to include wine in Skylab's rations), but won't listen to Morton Thiokol's engineers.

The Chernobyl accident was evidently the result of the local operators (and management?) ignoring the procedures in the operating manual; the Soviets claim that the local folks weren't supposed to have that much autonomy. The operators will take the rap--but the Soviet central management is responsible for not doing a better job of supervising (and motivating?) the local site people.

Right now, most of TVA's nuclear capacity is shut down; it seems that their plants don't match their documentation, due to unrecorded (and perhaps unauthorized) modifications during construction. Since this problem (at least at this magnitude) is unique to TVA, it seems that the fault was management's attitude towards the importance of this documentation. At least the NRC seems to think so, since a management reshuffle was one of their conditions for relicensing the TVA reactors.

No one's mentioned the earlier famous O-ring/management failure (so I have to, of course--): the triple engine failure on a 727. In this case, the ever-so-reliable O-rings failed because they were omitted from a maintenance kit--so they didn't get installed, so they didn't seal the engine chip detectors, so all the oil ran out of the engine, so all three engines failed en-route (over the ocean). One restarted (it still had a quart or so left), and the aircraft made it back to Miami. The NTSB decided the problem was inadequate training and supervision; the procedures for changing the O-rings had been changed, but no one told the mechanic, or checked his work (as they were required to do).

Hope this kills all further interest in O-rings--
--Martin Harriman Intel Santa Cruz <martin@srucad.sc.intel.com>

Re: Words, words, words...

Mike McLaughlin <mikemcl@nrl-csr>
Tue, 26 Aug 86 15:25:49 edt

[Herb's message got appended after the end of RISKS-3.42, and
was not included in the Contents of that issue. Sorry. PGN]

"Deliberate dumbness" is delightful. Reminds me of a friend's term, "malicious obedience," referring to carrying out dumb orders in their infinite complexity, regardless of the consequences, and without applying a grain of common sense.

Used car salesmen and realtors sometimes exhibit deliberate dumbness when they discourage the owner from telling them about defects in a property or automobile.

I do not know that "NO ONE in the scientific community believes that it is possible to frustrate a deliberate Soviet attack on the U.S. population..." If there is a PhD in a science who believes that, is that person de facto excluded from the scientific community?

I do not know what "frustrat(ing) a deliberate... attack" means. If it means deterring the attack by reducing the cost/benefit ratio to an unacceptable level, I believe that is possible (but I am not in the scientific community and never have been).

If it means saving a significant number of civilian lives from an inevitable attack, I believe that is possible (but...).

If it means saving EVERY civilian life, I do not believe that, any more than I believe the statement that "NO ONE... etc."

SDI involves more than science, it affects billions of people, millions of military and defense industry people, and thousands of decisions makers on both sides of the Curtain. As such, it is not susceptible to the simple and elegant solutions of science - neither "It won't work" nor "It will work" is adequate.

I have five children. I hope we, and the Russians, get it right, whatever we decide to do.

"Things are the way they are because if they were to be any different they
wouldn't have come out like this." - Tevye (Sholom Aleichem)

- Mike <mikemcl@nrl-csr.arpa>

Comments on paper desired

<LIN@XX.LCS.MIT.EDU>
Tue, 26 Aug 1986 19:42 EDT

I am currently writing a paper entitled COMPTER SOFTWARE AND STRATEGIC DEFENSE, which should be available in preliminary draft form on August 29, Friday. Comments are solicited by September 15. It is too big to mail, so FTP is the solution. If you want to see a copy (in exchange for a promise to make comments on it), please drop me a note. A brief abstract follows:

Computer software will be an integral part of any strategic defense system (defined here to include BMD, ASAT, and air defense). Several issues are addressed: The reliability of SDI software, the problem of system architecture, the problems that very short defensive time lines may introduce, the risk for accidental nuclear war, mechanisms for escalation control.

Thanks. Herb

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 44

Wednesday, 14 August 1986

Contents

- F-16 Problems
  Bill Janssen
- Various clips from European Newspapers
  Martin Minow
- Comment on Nancy Leveson's comment on...
  Alan Wexelblat
- Words, words, words...
  Herb Lin
- Software Safety
  Paul Anderson
- Info on RISKS (comp.risks)

F-16 Problems (from Usenet net.aviation)

Bill Janssen <janssen@mcc.com>
Wed, 27 Aug 86 14:31:45 CDT

A friend of mine who works for General Dynamics here in Ft. Worth wrote some of the code for the F-16, and he is always telling me about some neato-whiz-bang bug/feature they keep finding in the F-16:

o Since the F-16 is a fly-by-wire aircraft, the computer keeps the pilot from doing dumb things to himself. So if the pilot jerks hard over on the joystick, the computer will instruct the flight surfaces to make a nice and easy 4 or 5 G flip. But the plane can withstand a much higher flip than that. So when they were 'flying' the F-16 in simulation over the equator, the computer got confused and instantly flipped the plane over, killing the pilot [in simulation]. And since it can fly forever upside down, it would do so until it ran out of fuel.

(The remaining bugs were actually found while flying, rather than in simulation):

o One of the first things the Air Force test pilots tried on an early F-16

was to tell the computer to raise the landing gear while standing still on
the runway. Guess what happened? Scratch one F-16. (my friend says there
is a new subroutine in the code called 'wait_on_wheels' now...) [weight?]

o The computer system onboard has a weapons management system that will attempt to keep the plane flying level by dispersing weapons and empty fuel tanks in a balanced fashion. So if you ask to drop a bomb, the computer will figure out whether to drop a port or starboard bomb in order to keep the load even. One of the early problems with that was the fact that you could flip the plane over and the computer would gladly let you drop a bomb or fuel tank. It would drop, dent the wing, and then roll off.

There are some really remarkable things about the F-16. And some even more remarkable things in the new F-16C and D models:

o They are adding two movable vents called 'canards' that will be installed near the engine intake vent under where the pilot sits. By doing some fancy things with the flight surfaces and slick programming, they can get the F-16 to fly almost sideways through the air. Or flat turns (no banking!). Or fly level with the nose pointed 30 degrees down or up (handy for firing the guns at the ground or other aircraft).

I figured this stuff can't be too classified, since I heard the almost same thing from two different people who work at GD. I hope the Feds don't get too upset...

George Moore (gm@trsvax.UUCP)

<minow%regent.DEC@decwrl.DEC.COM>
27-Aug-1986 0835

(Martin Minow, DECtalk Engineering ML3-1/U47 223-9922)
To: risks@csl.sri.com
Subject: Various clips from European Newspapers

From The [London] Guardian, Aug. 20-22 1986 (not sure of the exact date):

Bank zaps `raid on computer'

Barclays Bank yesterday denied reports that computer experts had "hacked" into its Whitehall computer and transferred 440,000 Lb. Sterling to an overseas account.

From Dagens Nyheter [Stockholm], Aug. 22, 1986. My translation, abridged.

Shock billing of private person
Phone bill of 31,000 kronor [almost $2,600]

A woman in the Stockholm area received a record phone bill of 31,000 kronor. The amount is equivalent to local calls 24-hours per day for nearly two years.

The phone company's computers raised an alarm that the amount was unreasonably high, but human error resulted in the bill being sent out anyways. The group that normally checks especially high invoices never got to see this bill.

The woman and the phone company have reached an agreement, whereby she pays an average bill based on previous invoices. Phone technicians are now trying to discover whether an error occurred in the computer-controlled phone exchange. ...

"It's completely our fault," says phone company spokesman Kjell Palmqvist.

"What are you doing about it?" [asked the reporter.]

"First, we've come to an agreement with the woman. She need not pay more than normally. We've also started an examination of what could have caused the problem.... There could have been a problem in the computerized phone exchange, or a cable-error or other type of interference."

"Is this sort of bill common?"

"No, theoretically, we expect one error in 10,000 years. But no technology is 100% perfect." ...

The telephone exchange, in Oestermalm in Stockholm, uses an AXE-exchange, a computerized telephone exchange [manufactured by LM Ericsson] that is very advanced and reliable.

From Dagens Nyheter [Stockholm], Aug. 22, 1986. My translation, abridged.

Battle over Databank

The chairman of the governmental data- and public-access committee [offentlighetskommitt'en], Carl Axel Petri, rejects the criticisms which have recently been brought by the moderate party [conservative] and folk-party [liberal conservative] concerning sales of personal information from computer data banks.

[Sweden has a "sunshine" law, almost 200 years old, that guarantees public access to almost all government documents. As the information in the manual registers were considered public, so too is the same information in the computerised data bank. Information which is not public is carefully controlled. Access is governed by the Swedish Data Law, which is now over 10 years old.]

"It is important to quickly get a law that stops general sales. We have allowed some exceptions, nine specified computer companies, but even their sales shall, in the future, be controlled by parliament. Nobody should be allowed to earn money by [selling] personal

information. Sales should have a public interest, in principle, the
new law will forbid sales" said Petri. ...

The leader of the Moderate Party, Gunnar Hoekmark, says that Petri is incorrect when he claims that the law will forbid sales of personal information.

"On the contrary," says Hoekmark, "the largest databases will continue to be sold. Without the committee's discussing what effect sales of different personal information will have on individual personal integrity, they propose that the largest database, Spar, may continue to sell information on individuals income, personal identity number, wealth, civil status, address, age, etc."

Hoekmark points out that the majority [report?] of the inquiry didn't answer the most basic questions on whether the government in general shall have the right to sell information on private individuals' economy and personal situation.

The majority includes the Center Party's [liberal conservative] Olof Johansson, who says that the important issue for the future isn't whether the information ought to be sold, but what information should be collected. This includes, for example, the discussion on limitations of use of the personal id number.

Constitutional questions [the Sunshine Law is part of the Swedish Constitution] and the future of the personal id number will remain for the inquiry to solve by next spring.

Sloppily translated by Martin Minow

[Peter, I also have a long article on computer controlled airplanes (fly by wire) from the Observer. Mostly Sunday Paper background. Too much to type in. "... the pilot must have enough confidence in the flight control computer, and the men who programmed its software, to take off in an aircraft he cannot fly without them" "there is one more type of failure from which they [the pilots] cannot recover."]

Comment on Nancy Leveson's comment on...

Alan Wexelblat <wex@mcc.com>
Wed, 27 Aug 86 09:33:11 CDT

I agree in large part with Nancy Leveson's comments in RISKS-3.43. Nevertheless, I find it interesting that she denies that there are "human errors" but believes that there are "management errors." It seems that the latter is simply a subset of the former (at least, until we get computer managers). Also, it's not clear whether she includes things like `pushing the wrong button' or `following the wrong procedure' under the category of "operational errors."

--Alan Wexelblat (WEX@MCC.COM)

Words, words, words...

<LIN@XX.LCS.MIT.EDU>
Wed, 27 Aug 1986 15:05 EDT

> From: mikemcl at nrl-csr (Mike McLaughlin)
>
> I do not know that "NO ONE in the scientific community believes that it is possible to frustrate a deliberate Soviet attack on the U.S. population..." If there is a PhD in a science who believes that, is that person de facto excluded from the scientific community?

I should have been more precise. No person with technical credentials has stated that it is possible to deny the Soviet Union the capability to wreak significant damage on the U.S. population and industry.

> I do not know what "frustrat[ing] a deliberate... attack" means.
>
> If it means deterring the attack by reducing the cost/benefit ratio to an unacceptable level, I believe that is possible (but I am not in the scientific community and never have been).
>
> If it means saving a significant number of civilian lives from an inevitable attack, I believe that is possible (but...).

I think the benchmark that Ashton Carter used in his Office of Technology Assessment background paper on BMD was pretty good, and it will serve as a starting point for discussion. "Frustrate a deliberate attack..." is taken to mean "preventing the Soviet Union from delivering by ballistic missile 100 megatons of nuclear warhead on U.S. cities and industry." (Note well: WW II was a 5 MT war.)

Software Safety

Paul Anderson <anderson>
Wed, 27 Aug 86 09:43:03 edt

I have received a copy of a proposed revision of MIL-STD-882B (System Safety Hazard Analysis) Task 212, Software Safety Analysis, that has been distributed for formal coordination. This task will be invoked on contractors building systems containing software for DOD. This task will require the contractor to conduct safety analyses and testing of the software, both on the software alone, and when integrated with the overall system.

If anybody has thoughts, comments, or suggestions (or even recommended wording), on what should be included in this task, please let me know (preferably within the next week or so).

Paul Anderson
anderson@nrl-csr

Search RISKS using swish-e

Report problems with the web pages to the maintainer

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 45

Thursday, 28 August 1986

Contents

- Nonviolent Resistor Destroys Aries Launch
  PGN
- Risks in the design of civil engineering projects
  Annette Bauman
- ATMs
  Lindsay F. Marshall
- Re: Typing Profiles
  Lindsay F. Marshall
- Human errors prevail
  Ken Dymond
  Nancy Leveson
- Info on RISKS (comp.risks)

Nonviolent Resistor Destroys Aries Launch

Peter G. Neumann <Neumann@CSL.SRI.COM>
Thu 28 Aug 86 21:30:48-PDT

From SF Chronicle wire services, 28 Aug 1986: White Sands Missile Range NM

A rocket carrying a scientific payload for NASA was destroyed 50 seconds after launch because its guidance system failed... The loss of the $1.5 million rocket was caused by a mistake in the installation of a ... resistor of the wrong size in the guidance system. "It was an honest error", said Warren Gurkin... "This rocket has been a good rocket, and we continue to have a lot of faith in it." Saturday's flight was the 27th since the first Aries was launched in 1973, and it was the third failure.

Risks in the design of civil engineering projects

<ABauman @ DDN1>
28 Aug 86 06:40 EDT

Computer-Aided Engineering, Penton Publishing, Cleveland OH, April 1986 page 4:

"Impressive computer analysis, however, may tempt some engineers into developing designs that barely exceed maximum expected operational loads. In these cases there is no room for error, no allowance for slight miscalculations, no tolerance for inaccuracy. In engineering parlance, the design is "close to the line". The reasoning, of course, is that relatively small safety factors are justified because computer analysis is so accurate.
The major flaw in this logic, however, lies in the fact that the initial mathematical model set up by the designer may itself contain gross inaccuracies... These errors are carried through the entire analysis by thecomputer, of course, which uses the model as the sole basis for its calculations... And wrong answers are easily obsuured by flashy color graphics and high-speed interactive displays. In most cases, the engineer must be extreamly familar with the design and the programs used in its development to spot errors in results." -John K. Krouse editor

Annette C. Bauman, DDN-PMO
Test & Evaluation Branch, DDN Network software Test Director

Re: ATMs

"Lindsay F. Marshall" <lindsay%kelpie.newcastle.ac.uk@Cs.Ucl.AC.UK>
Wed, 27 Aug 86 08:38:38 bst

>....Their dispensing machines cannot be cheated in this way, because they have
>a steel door in front of the machine which does not open until you insert a
>valid plastic card.

People who swindle ATM's don't have cash cards?????

ATM swindle's don't seem to have caught on in the UK too much yet (at least not that I've heard), but the new "vandal proof" phone boxes which have special money compartments seem to be rather more vulnerable. I have heard reports of people touring regions of the UK on a regular basis emptying these phones. Another interesting scam at the moment (which I presume has swept the US long ago....) and which is not illegal is that of beating quiz machines. Teams of 3 "experts" (sport, TV/film and general knowledge usually) tour pubs and play the video quiz machines. These have money prizes and they simply strip them of everything in them by answering all the questions. Most landlords are now removing these games as they are losing money.......

Re: Typing Profiles

"Lindsay F. Marshall" <lindsay%kelpie.newcastle.ac.uk@Cs.Ucl.AC.UK>
Wed, 27 Aug 86 08:29:32 bst

John Ellenby (of Grid systems) told me that they installed just such a thing into an operating system they were building and used it to distinguish between the various operators who used the console. The operators never could work out how the system "knew" who they were. (I may say that I am not totally convinced however - particularly in a non-keyboard oriented society such as the UK where very few people can actually type properly.)

Human errors prevail -- Comment on Nancy's Comment on ...

"DYMOND, KEN" <dymond@nbs-vms.ARPA>
28 Aug 86 14:11:00 EDT

Nancy Leveson's comment (on PGN's comment on human error in RISKS-3.43) makes some very good points. We do need to discuss the terms we use to describe the various ways systems fail if only because system safety and especially software safety are fairly young fields. And it seems natural for practitioners of a science, young or not, to disagree on what they are talking about. (Recall the discussion a few years ago in SEN on what the term "software engineering" meant and whether what software engineers did was really engineering.)

But what scientists say in these discussions about science may not be science, at least in the sense of experimental science -- it's more like philosophy, especially when the talk is about "causes". Aristotle, for one, talked a lot about causes and categories. When we are urged to constrain our use of "cause" ("Trying to simplify and ascribe accidents to one cause will ALWAYS be misleading. Worse, it leads us to think that by eliminating one cause, we have then done everything necessary to eliminate accidents (e.g. train the operators better, replace the operators by computer, etc.)"), we are being given a prescription, something value-laden. (I don't mean to imply that science is or should be value-free.) The implication in the prescription seems to be that we (those interested in software and system safety) should avoid using "cause" in a certain way otherwise we are in danger of seducing ourselves as well as everybody else not specifically so interested (the public) into a dangerous (unsafe) way of thinking.

But a way of supplementing the philosophical or prescriptive bent to our discussion about the fundamental words is to look at how other disciplines use the same words. For example structural engineers seem to be doing a lot of thinking about what we would call safety. They even say "Human error is the major cause of structural failures." (Nowak and Carr, "Classification of Human Errors," in Structural Safety Studies, American Society of Civil Engineers, 1985.) It may be that our discussions about the basic words we use can be helped by consulting similar areas in more traditional types of engineering.

There is another prescriptive aspect to the subject of constraining our discourse as raised by Nancy, namely not admitting into that discourse statements from certain sources. ("Also, the nature of the mass media, such as newspapers, is to simplify. This is one of the dangers of just quoting newspaper articles about computer-related incidents, When one reads accident investigation reports by government agencies, the picture is always more complicated.") Our thinking about this prescription may also benefit from looking at other engineering disciplines to see how they investigate and report on failures and what criteria and categories (the jargon word is "methodology") they use, implicitly or explicitly, in assigning causes to failure. "Over-simplified" might be the best adjective to describe some of the contributions to RISKS from newspapers-- one doesn't know whether to believe them or not. A problem may arise when writers on safety start to quote SEN and the safety material collected there, most of which is previewed here on RISKS, as authoritative sources on computer and other types of failures. The question is whether SEN's credibility is being lessened or the newspaper's enhanced by the one being the source for the other. Compare some of the newspaper stories reproduced on this list with the lucidity and thoroughness of Garman's report on the "The 'Bug' Heard 'Round the World," (SEN, Oct. 1981). That seems a model for a software engineering analysis and report of a failure. We might compare it to other thorough engineering analyses of failures, say the various commissions' reports on Three Mile Island or the NBS (no chauvinism intended) report on the skywalk collapse at the Hyatt Regency in Kansas City. (The report of the Soviet government on Chernobyl will perhaps bear reading, too.)

If we evolve some kind of standard for analyzing and reporting system failure, we'll be able to categorize the trustworthiness of newspaper and, for that matter, any other failure reports so that their appearance on RISKS will not necessarily count as an endorsement, either in our own minds or in that of the public.

Ken Dymond, NBS

Human errors prevail -- Comment on Alan Wexelblat's Comment on

Nancy Leveson <nancy@ICSD.UCI.EDU>
28 Aug 86 19:42:14 PDT (Thu)

Nancy Leveson's... (ad infinitum?) [but not quite yet ad nauseum!]

From Alan Wexelblat's comment on my comment on ... (RISKS-3.44):

>... she denies that there are "human errors" but believes that
>there are "management errors." It seems that the latter is simply
>a subset of the former (at least until we get computer managers).

With some risk of belaboring a somewhat insignificant point, after reading [Alan's message], it is clear to me that I did not make myself very clear. So let me try again to make a more coherent statement. I did not mean to deny that there are human errors, in fact, the problem is that all "errors" are human errors.

I divide the world of things that can go wrong into human errors and random hardware failures (or "acts of God" in the words of the insurance companies). My real quibble is with the term "computer errors". Since I do not believe that computers can perform acts of volition (they tend to slavishly and often frustratingly follow directions to my frequent chagrin), erroneous actions on the part of computers must either stem from errors made by programmers and/or software engineers (who, for the most part, are humans despite rumors to the contrary) or from underlying hardware failures or a combination of both. I suppose we could also include operator errors such as "pushing the wrong button" or "following the wrong procedure" as either part of "computer errors" or as a separate category. The point is that the term "computer error" includes everything (or nothing depending on how you want to argue) and the term "human error" includes most everything and overlaps with most of the computer errors. And the term "computer error" is also misleading since to me (and apparently to others since they tend to talk about human errors vs. computer errors and to imply that we will get rid of human errors by replacing humans with computers) it seems to imply some sort of volition on the part of the computer as if it were acting on its own, without any human influence, to do these terrible things.

That is why I do not find the terms particularly useful in terms of diagnosing the cause of accidents or devising preventative measures. I was just trying to suggest a breakdown of these terms into more useful subcategories, not to deny that there are "human errors" (in fact, just the opposite). And in fact, to be useful, we probably need to further understand and subdivide my four or five categories which included design flaws, random hardware failures, operational errors, and management errors (along with the possibility of including production or manufacturing errors for hardware components). Note that three out of the first four of these are definitely human errors and manufacturing errors could be either human-caused (most likely) or random.

Actually, I thought the part of my original comment that followed the quibbling about terms was much more interesting...

Nancy Leveson
ICS Dept.
University of California, Irvine

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 46

Saturday, 30 August 1986

Contents

- Human error
  - Nancy Leveson
  - Lindsay F. Marshall
- Re: F-16 Tales
  - Earl Boebert
  - Phil Ngai
- Correction to note about flight simulators
  - Martin Minow
- Supermarket grinds to a halt
  - David Sherman
- Video processing
  - Guy Schafer
- ATMs
  - Jacob Palme
- Info on RISKS (comp.risks)

Human error

<LIN@XX.LCS.MIT.EDU>
Fri, 29 Aug 1986 18:48 EDT

From: Nancy Leveson

Re: Human Error

"Lindsay F. Marshall" <lindsay%kelpie.newcastle.ac.uk@Cs.Ucl.AC.UK>
Fri, 29 Aug 86 12:08:04 bst

Someone who has looked at the changing attitudes to "human error" as against "mechanical failure" is Michael Lesk. He has been studying reports of railway accidents in the UK to extract from them information about the attitudes of the reporters and investigators towards the causes of the accidents. I don't know if he has written this up anywhere or not, nor do I know if he reads RISKS. He is well worth talking to

about the subject however and has uncovered some exceedingly
interesting points.
Lindsay F. Marshall

[Will someone at Bell Labs who reads this please give Mike a nudge? PGN]

Re: F-16 Tales

<Boebert@HI-MULTICS.ARPA>
Fri, 29 Aug 86 10:51 CDT

Weight on wheels is a basic sensor input that tells the flight program whether or not the aircraft is airborne. In advanced systems like the F-16 its is probably confirmed by air data computer and inertial platform inputs; in older systems, where the computer does just nav and weapons delivery, it is the prime indicator. It is therefore unlikely in the extreme that this would be overlooked in a design or an ordnance safety analysis ((weight_on_wheels = TRUE) & (master_arm = TRUE) & (weapon_release = TRUE) is clearly an undesired state). I am also skeptical that the gear would be controlled by the flight computer, but I am not familiar with the F-16 so cannot comment further.

F-16 software

Phil Ngai <amdcad!phil@decwrl.DEC.COM>
Fri, 29 Aug 86 19:57:30 pdt

It sounds very funny that the software would let you drop a bomb on the wing while in inverted flight but is it really important to prevent this? Is it worth the chance of introducing a new bug to fix this very minor problem? Is it worth the chance of making the code too big to fit in memory? What is the chance that a pilot would really make this mistake?

[The probability is clearly NONZERO. It is very dangerous to start making assumptions in programming about being able to leave out an exception condition simply because you think it cannot arise. Such assumptions have a nasty habit of interacting with other assumptions or propagating. PGN]

Correction to note about flight simulators

Martin Minow, DECtalk Engineering ML3-1/U47 223-9922 <minow%regent.DEC@decwrl.DEC.COM>
29-Aug-1986 1406

In a private mail exchange, Danny Cohen ("COHEN@B.ISI.EDU") was kind enough to point out that I had mis-remembered my article from Smithsonian where I claimed the article stated that a flight instructor flew as a flight engineer on a commercial flight.

> The plane encountered a wind-shear situation on take off. The
> instructor, from his flight engineer's position, reminded the pilot
> that the correct recovery for wind-shear is opposite to the correct
> recovery for a stall (which has a similar appearance to the pilot)."

According to Danny (I can't find my copy of this issue), the article does not talk about anything being "opposite to the correct recovery for the stall."

I'm sorry for the confusion this might have caused anyone. At least, I did learn a lot about flying and recovery from dangerous conditions. Danny did ask me to clarify my purpose in submitting the article to RISKS -- whether it was to show that computer-based simulators contribute to airline safety, or to "highlight the risks in using computers for whatever purposes." To set the matter straight, it was to show that computer-based simulation is a factor in increased airline safety, as it lets pilots learn about situations that are either dangerous or unusual (or both) in real life.

Danny is still looking for pointers to accidents caused by computer-based simulators.

Martin

[I don't mean to take a potshot at Martin, who has been a delightful contributor. But PLEASE, all of you, if you see something that you think is appropriate for RISKS, make a note of it at the time rather than subsequently half-remember it. I keep a huge stack of old items next to my terminal just in case I have to dig back... PGN]

Supermarket grinds to a halt

<mnetor!lsuc!dave@seismo.CSS.GOV>
Fri, 29 Aug 86 17:04:53 edt

Last week I went to our local Miracle Food Mart supermarket (in northern Toronto) at 9 a.m. on a Sunday, when they were just opening. They discovered that they couldn't get any of the cash registers to work; something was down in the central system. So they had the cashiers writing each number down on a pad of paper and totalling them up by hand, which slowed checkout down to a crawl. After a while, someone found a desk calculator with a paper tape, which made things a bit faster. When I left they had someone at the door warning customers not to bother coming in because the terminals weren't working.

Obviously, this kind of thing can happen only where cash registers are no longer cash registers but terminals connected to a central system, which is becoming more and more the case. I can't believe MFM doesn't have some type of backup system, since they're a large chain. My speculation is that someone wasn't prepared for the system to be running on Sunday morning; supermarkets must be closed in Ontario on Sundays, and the ones near us started opening only about a month ago...

David Sherman, The Law Society of Upper Canada, Toronto

{ ihnp4!utzoo seismo!mnetor utzoo hcr decvax!utcsri } !lsuc!dave

---

Video processing

Guy Schafer <decwrl!amdcad!amdimage!prls!philabs!linus!axiom!gts@ucbvax.Berkeley.EDU>
Thu, 28 Aug 86 15:38:31 edt

Now that sophisticated hardware for capturing and altering video images exists for even the modest IBM-PC (AT&T's Truevision products), several concerns arise:

Because images can be captured in real time (for less than $5000), and it has been proven that at least one method exists for over-powering ('hi-jacking') a cable video broadcast, some program can be altered and re-broadcast (with a delay equal to the video processing time). This could be especially dangerous if it is done to, say, 2 minutes of a news broadcast or televised political proceedings.

Video post-processing can also be an effective means to control the behavior of an individual by tapping directly into her cable coming into her house. An appropriate stock tip given by a seemingly authentic Ruekeiser (sp?) might cause a major stockholder to get on her phone to a broker with predictable (and thus profitable) results. We always knew a hacker with a PC had quite a bit of power; if this hacker can alter someone's main source of information (television broadcasts) he suddenly has quite a bit more.

Also, video tapes which are used as evidence in court can be changed without simple means of detection.

Actors can be cheated out of royalties--especially in commercials where post-processing 30 seconds of video could cost less than the royalties of an often-repeated performance. The features of the face can be "airbrushed" or distinguishing marks can be added or removed (by software--e.g. TIPS by AT&T) and the actor told that someone else got the part.

Any comments?

>< ...{ decvax!linus | seismo!harvard }!axiom!gts

ATMs (RISKS-3.45)

<Jacob_Palme_QZ%QZCOM.MAILNET@MIT-MULTICS.ARPA>
30 Aug 86 16:57 +0200

<>..Their dispensing machines cannot be cheated in this way, because they have
<>a steel door in front of the machine which does not open until you insert a
<>valid plastic card.
>
>People who swindle ATM's don't have cash cards?????

If you have a legally obtained cash card, and insert it into the

machine, this act is immediately recorded, so that if the swindle
is detected, they can find out who did it.

If you have an illegally obtained cash card, you probably do not
know the password you have to input on the keyboard. When you
input the wrong password (or do not input any password at all),
the machine swallows the card, and you never get it back again.

At least that is the way the Swedish ATM's work.

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 47

Monday, 1 September 1986

Contents

- Flight Simulators Have Faults
  Dave Benson
- Re: QA on nuclear power plants, the shuttle, and beer
  Henry Spencer
- Acts of God vs. Acts of Man
  Nancy Leveson -- two messages
- Computer Literacy
  Mike McLaughlin
- Another supermarket crash
  Ted Lee
- A supermarket does not grind to a halt
  Brint Cooper
- Info on RISKS (comp.risks)

Flight Simulators Have Faults

Dave Benson <benson%wsu.csnet@CSNET-RELAY.ARPA>
Sat, 30 Aug 86 23:08:47 pdt

I mentioned the F-16 RISKS contributions to my Software Engineering class yesterday. After class, one of the students told me the following story about the B-1 Flight Simulator. The student had been employed over the summer to work on that project, thus having first-hand knowledge of the incident.

Seems when a pilot attempts to loop the B-1 Flight Simulator that the (simulated) sky disappears. Why? Well, the simulated aircraft pitch angle was translated by the software into a visual image by taking the trigonometric tangent somewhere in the code. With the simulated aircraft on its nose, the angle is 90 degrees and the tangent routine just couldn't manage the infinities involved. As I understand the story, the monitors projecting the window view went blank.

Ah, me. The B-1 is the first aircraft with the capability to loop? Nope, its been done for about 70 years now... The B-1 Flight Simulator is the first flight simulator with the capability to allow loops? Nope, seems to me I've played with a commercially available Apple IIe program in which a capable player could loop the simulated Cessna 180. $$ to donuts that military flight simulators with all functionality in software have been allowing simulated loops for many years now.

Dick Hamming said something to the effect that while physicists stand on one another's shoulders, computer scientists stand on one another's toes. At least on the toes is better than this failure to do as well as a game program... Maybe software engineers dig one another's graves?

And this company wants to research Starwars software... Jus' beam me up, Scotty, there's no intelligent life here.

Re: QA on nuclear power plants, the shuttle, and beer

<decwrl!decvax!LOCAL!utzoo!henry@ucbvax.Berkeley.EDU>
Sun, 31 Aug 86 01:35:29 edt

Equipment failures and human errors are common enough in any human endeavor; the question is not whether they happen, but whether they present actual or potential risks of serious consequences. In this context the lack of publicity is not at all surprising: one form of serious consequence is public hysteria over insignificant trivia. When an attempt to reach a vacationing brewery programmer gets blown up into stories of a total production shutdown and impending beer shortage -- this, mind you, in an industry which is *not* the focus of hostile propaganda campaigns and widespread irrational fears -- the people involved with nuclear plants have every reason to be very quiet about even routine, unexciting, non-hazardous problems.

Henry Spencer @ U of Toronto Zoology
{allegra,ihnp4,decvax,pyramid}!utzoo!henry

Acts of God vs. Acts of Man

Nancy Leveson <nancy@ICSD.UCI.EDU>
30 Aug 86 17:26:20 PDT (Sat)

<> From: Nancy Leveson

Acts of God vs. Acts of Man, Round n+1 (eastbound)

Nancy Leveson <nancy@ICSD.UCI.EDU>
30 Aug 86 23:30:36 PDT (Sat)

>From Herb Lin:
>But the number of assumptions that
>designers must make is enormously large, and it is essentially
>impossible to even articulate ALL of one's assumptions.

Agreed. But there are ways to determine which are the critical assumptions with regard to particular hazards. This is exactly what some of my techniques, e.g. software fault tree analysis, attempt to do. In the Firewheel example that I published, we determined a critical assumption which could have resulted in the satellite being destroyed. That is, if there were two sun pulses detected within 64 milliseconds of each other, the microprocessor interrupt system became hung which could possibly result in destruction of the sensor booms (and thus the usefulness of the satellite). We found this assumption by working backward through the software from the hazardous condition. The solution, once the critical assumption had been determined, was a simple blocking of the second sun pulse interrupt.

I don't know for what size systems these backward analysis approaches are practical. It took Peter Harvey (my student) two days to analyze the Firewheel software (which is about 1600 lines long) by hand. Obviously, it would be possible to analyze larger software, but we do not yet know how much this will scale up practically. We are working on a software tool to automate as much as possible. These techniques are, of course, no more perfect than other more traditional software engineering techniques. And better ones may be found. I am just not ready to say it is impossible without first trying.

Backwards analysis, verification of safety, software interlocks, software fault tolerance, fail-safe design, ... -- there are possible solutions which we should be examining.

Nancy Leveson

Computer Literacy

Mike McLaughlin <mikemcl@nrl-csr>
Mon, 1 Sep 86 11:49:10 edt

From THE WASHINGTON POST, Monday, 1 Sept 86, page A14, Letters to the Editor ["..." indicates omissions]. While I do not entirely agree with Mr. Jordan, much of what he says is directly applicable to Risks.

> [Before responding to this, please recall that this topic has already been discussed at some length in RISKS-2.36 and 37, and in RISKS-3.17, 19, 20, and 21. PGN]

++

COMPUTER LITERACY

Although I earn my living as a consultant in computerized data bases, I strongly oppose the view... that computer literacy should be mandatory in the secondary school curriculum.

Computers are a device for performing some task that either is already performed by other means or first must be understood in other terms, usually a mathematical equation. Learning how to operate a computer, or program one, is not going to improve a student's knowledge of languages, mathematics, history or political science.

Alfred North Whitehead observed that civilization increases the number of things that we can do without thinking, i.e., that we can take for granted. This is evident in the development of computers, which increasingly are becoming like automobiles; anyone can drive them. Learning the technology of computers has as much relevance to everyday life as learning the technology of auto engines.

Unquestionably there are tasks for which computers are indespensable, but individuals will learn those functions as they become involved in the task itself, whether it be medical diagnosis, controlling the flow of electric power over a grid or determining the authorship of a 16th-century poem.

What students need to know is how to think, especially about the human condition. As more and more college students flock to "practical" majors, the secondary schools should be concentrating on the liberal arts. In this perspective, "computer literacy" may be just another form of a larger illiteracy.

- John S. Jordan, Washington, D.C.

Another supermarket crash

<TMPLee@DOCKMASTER.ARPA>
Sat, 30 Aug 86 23:26 EDT

Same thing happened here (Minneapolis-St.Paul) a couple of years ago -- I was in a major discount store (Target), during a normal busy time -- Saturday morning, I think -- only to find that all the cash registers wre down because the central computer was down. Don't know how long it lasted, but at least long enough that by the time I got there the cashiers were using paper and pencil. Really, stupid -- (so he says as a so-called computer expert) -- given that those registers probably had Z80's or 6502 or such in them, a printed record, etc., so they could just as well have worked off-line (except for not knowing the price on instant sale items, which would I assume have been in the central machine.)

A supermarket does not grind to a halt

Brint Cooper <abc@BRL.ARPA>
Mon, 1 Sep 86 13:29:45 EDT

Last month, as I awaited checkout in a "Giant" supermarket in Bel Air, MD, an area-wide power outage lasting several minutes occured. The following was the sequence of events:

1. All lights, outside and inside, instantly out.
1A. Display on cash register_cum_terminal RETAINED display!

2. Some of the same lights came back almost immediately (seemed to be back-up power).
3. Several minutes passed; additional lights began to come on.
4. One-by-one, the terminals "beeped" and became functional.

No market employee with whom I spoke seemed to understand what actually happened. But the computer system obviously was protected from such random power outages (which occur FREQUENTLY here).

Brint Cooper

UUCP: ...{seismo,unc,decvax,cbosgd}!brl-smoke!abc

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 48

Tuesday, 2 September 1986

Contents

- Aeromexico Crash
  UPI via PGN
- Air Force puts secrets up for sale
  Peter G. Neumann
- Randi, Popoff, and Data Privacy Laws
  Phil Karn via Geoff Goodfellow
- Flight Simulators Have Faults
  Gary Whisenhunt
- On-Line with Taco Bell Telephone
  John Mulhollen
- Titanic photo expedition
  Lindsay F. Marshall
- New Zealand $1 million deposit
  Dave Sherman
- Examination Processing Error
  Joe Stoy
- Info on RISKS (comp.risks)

Aeromexico Crash

Peter G. Neumann <Neumann@CSL.SRI.COM>
Tue 2 Sep 86 09:59:20-PDT

The New York Times news summary, Tuesday, 2 Sept 1986, had this item on the LA plane crash.

New York - The California plane collision Sunday occurred in a government-established restricted zone where the private plane that was destroyed in the collision with an Aeromexico DC-9 was not authorized to fly, the Federal Aviation Administration said. An FAA spokesman also said the controller guiding the DC-9 could not have radioed warnings to avert the collision because ''as far as we can determine'' no radar blip designating the small plane appeared on his scope. The controller did not know of the small plane's existence, the spokesman said.

A SF Chron report on the same day indicated that the controller in question was distracted by the pilot of another private plane, with whom he was having a two-minute interaction -- during which time the crash occurred.

PBS added several more pieces to the puzzle. The pilot of the private plane (a Piper Archer) apparently had had a heart attack just before the crash. The private plane did indeed appear on the controller's radar after all. However, it was not equipped with an altitude-measuring transponder, so the controller had no idea whether or not there was any danger.

The death toll is 64 on the jetliner, 3 on the Piper PA-28, and at least 18 on the ground.

Air Force puts secrets up for sale

Peter G. Neumann <Neumann@CSL.SRI.COM>
Tue 2 Sep 86 16:00:31-PDT

Fred Ostapik went off to Ashland, Oregon, for some Shakespeare plays, and brought back this clipping from the local Ashland paper of 23 August 1986:

Audit: Air Force put secrets up for sale

Washington (UPI) -- A military audit, examining the latest lapse in Pentagon security, says the Air Force inadvertently allowed computer tapes containing ``sensitive, unclassified'' data to be auctioned off to the public.

The Air Force Audit Agency found more than 1,200 magnetic tapes containing the data -- dealing with launch times, aircraft tests, and launch and aircraft vehicles -- available for public purchase at three key bases...

Auditors said they found 1,980 analog tapes available for purchase, 64 percent of which had not been erased and contained sensitive unclassified data. Five of the seven installations checked had inadvertently made secret tapes available to the public.

Randi, Popoff, and Data Privacy Laws

Phil Karn <karn@ka9q.bellcore.COM>
31 Aug 86 02:29:11 GMT

I picked up a copy of the magazine "Free Inquiry" at the bookstore today. The cover article was written by James Randi (the magician who debunks lots of ESP frauds). In fact, the magazine seems to be run by the same folks who do the Skeptical Inquirer, but is slanted more towards religious debunking.

Randi's article was titled "Peter Popoff Reaches Heaven via 39.17 Megahertz". Popoff is one of the most notorious TV faith healers. Randi's group went to the shows and noticed that Popoff wore a hearing aid. Then they got a scanner and quickly found the frequency his wife was using to tell him the names and ills of people whom she had pumped for information before the show.

Now ponder the fact that the proposed Communications Privacy Act now pending in the US Senate would have made this expose' illegal. The conversation was meant to be private, and Popoff certainly would have objected to its interception.

Could there be a connection here? Hmm......

Phil

Flight Simulators Have Faults

Gary Whisenhunt <gwhisen%ccvaxa@GSWD-VMS.ARPA>
Tue, 2 Sep 86 10:35:47 cdt

I developed flight simulators for over 7 years and could describe many such bizarre incidents. I seriously doubt that the sky went blank in the B-1 simulator when it was delivered to the government. Military simulators have formal acceptance tests that last for months. The last one that I worked on had a test procedure over 12 inches thick. To point out a failure during testing (or more likely development) seems meaningless. Failures that make it into the actual product are what should be of concern.

Most flight simulators procured by the Air Force and the Navy require Mil-Std 1644 or Mil-Std 1679 to be followed when developing software. These standards detail how software is to be developed and tested. The standards are fairly strict and exhaustive. This is to ensure product correctness even if it incurrs greater costs. It would be interesting study for a class in Software Engineering.

The greatest risks that I see from flight simulators (especially military) is that the simulator often lags behind the aircraft in functionality by a year or 2. Simulators require design data to be frozen at a certain date so that the simulator can be designed using consistent, tested data. After 2 years of development, the aircraft may have changed functionaly (sometimes in subtle ways) from the simulator design. The effect is much more dramatic for newer aircraft than it is for more established ones. The simulator is upgraded, but during the upgrade period pilots train on a simulator that is mildly different from their aircraft.

As for the effectiveness of simulators, I've been told by more than one pilot that the simulator saved his life because he was able to practice malfunction conditions in the simulator that prepared him for a real emergency that occurred later.

Gary Whisenhunt
Gould Computer Systems Division
Urbana, Ill.

[I thought that by now these simulators were designed so that they could be driven by the same software that is used in the live aircraft -- a change in one place would be reflected by the same change in the other, although changing the application code without having to modify the simulator itself. Maybe not... PGN]

On-Line with Taco Bell Telephone

John Mulhollen <JOHNM@USC-ECLC.ARPA>
Mon 1 Sep 86 22:32:00-PDT

It seems that more and more fast food places are switching from the old-fashioned cash register to computerized ones that enable management to get reports on how many burgers we sold today between 10pm and 11pm, the average number of tacos per patron, or how many french fries were wasted.
[Results are automatically telecommunicated back to headquarters. PGN]
However, along with the capability for better-informed management, the capability for unbelievable confusion also increases. Case in point -- our local Taco Bell has been "computerized" for almost 9 months now (equipment from Par Microsystems in NY) and patrons and employees alike have become accustomed to not getting receipts, and other quirks. Last week, the computer "locked up" (their term) just as I arrived. It was also just before the noon rush. The employees behind the counter did not know what to do. Do we take orders (on paper) and wait for the machine to come back up? Do we tell the customers to go away? It appears that with all this wonderful automation, the employees were incapable of 1) figuring out what to do; 2) taking orders without the computer; and 3) figuring out not only the total due for each patron, but the amount of change to return!!

When I was working my way through school, I did a brief stint at a local taco joint. We had an "old-fashioned" cash register (it didn't even compute the change -- how backward can you get!!) and we did just fine. When it didn't work, we just used a pad of paper (we knew all the prices and such).

Apparently one of the risks to society of the increasingly wide-spread use of computers is the possibility of losing the ability to think and reason.

JohnM

Titanic photo expedition

"Lindsay F. Marshall" <lindsay%cheviot.newcastle.ac.uk@Cs.Ucl.AC.UK>
Mon, 1 Sep 86 09:10:44 gmt

There was a program last night on ITV about the Woods Hole expedition to the Titanic. During the first dive, the program that was being used to help locate the ship "developed a mind of its own" and the people on the support ship had to guess headings for the sub to follow. Does any one have information on this??

Lindsay

New Zealand $1 million deposit (RISKS-3.41)

<mnetor!lsuc!dave@seismo.CSS.GOV>

Tue, 2 Sep 86 14:22:27 edt

>Bank machine is no match for schoolboy with a lollipop
>
> AUCKLAND, New Zealand [UPI] -- A schoolboy outsmarted an automatic
>bank machine by using the cardboard from a lollipop packet to
>transfer $1 million New Zealand dollars into his account, bank
>spokesmen said Thursday.

As the article indicates, this wasn't caught because of delays in reconciling the physical deposits with the computer records (4 WEEKS? my bank does it in a day!).

I find it somewhat misleading and irritating that the media choose to make a big deal about the lollipop packet. Obviously, he could have fed in an empty envelope just as easily. But "outsmarted ... by using the cardboard from..."? I guess this is one of the RISKs of having reporters who feel they need to make their stories interesting.

Dave Sherman, The Law Society of Upper Canada, Toronto
{ ihnp4!utzoo seismo!mnetor utzoo hcr decvax!utcsri } !lsuc!dave

---

Examination Processing Error

Joe Stoy <stoy%sevax.prg.oxford.ac.uk@Cs.Ucl.AC.UK>
Mon, 1 Sep 86 13:56:43 GMT

EXAMINATION PROCESSING ERROR

The following is copied (without permission) from The Times (London).
(C) TIMES NEWSPAPERS LIMITED 1986.

[Glossary:
O level ("Ordinary level") - an exam. taken by children aged fifteen or so.
A level ("Advanced level") - an exam. taken two years after O level; a
 prerequisite for university entrance.
CSE ("Certificate of Secondary Education") - an exam. for children who are not
 up to O level standard.
GCSE ("General Certificate of Secondary Education") - a forthcoming
 amalgamation of O level and CSE, in preparation for which some boards are
 already setting papers common to both existing exams.]
[[American readers should note that Public School means Private School. PGN]]

[28 August 1986]
COMPUTER MARK STARTS O-LEVEL PANIC
By Lucy Hodges
Education Correspondent

Hundreds of pupils who took a new joint O level/CSE examination in chemistry received the wrong grade because of a computer error.

It meant that no candidate received more than a grade C, the pass mark at O level, sending many parents and their offspring into a panic.

Schools were telephoned to be asked if this meant that the pupils involved would be prevented from doing chemistry at A level next year. The schools queried the grades with the boards and the rogue computer program was discovered.

The examination boards involved are the three GCE boards, Cambridge, Oxford and Cambridge, Southern Universities Joint, and the two CSE boards, West and East Midlands.

These five boards are combining to form the Midlands Examining Group for the new GCSE exam. As part of their preparation they are running joint examinations in certain subjects and new computer programs have had to be set up.

"The boards have to collaborate and with new computer programs we cannot find out mistakes until something happens," Mr. John Reddaway, secretary of the Cambridge board, said.

A total of 12,000 students entered for the joint examination in chemistry, of which 3,800 were awarded a grade C by the computer. In fact 800 of these should have been a grade A and 1,000 a grade B, Mr. Reddaway said.

The error appears to have occurred at the offices of the West Midlands CSE board in Birmingham, which was administering this particular exam. Mr. Reddaway said that the mistaken grades had all been rectified. "I hope schools and colleges will receive them tomorrow."

Whitgift School in Croydon, a boys' public school which normally gets very good results, was one of those involved. It was surprised to find that all its O-level pupils had been awarded a grade C.

"It was ridiculous in a school like this not to have any grades A or B," Miss Patricia Dawson-Taylor, the school secretary, said. "I told the board that we would be querying them."

Parents of Whitgift boys have been informed by the school that there has been an error and that some candidates may be upgraded.

[29 August 1986 -- excerpts from the follow-up report]

EXAMS RESULT IS CORRECTED

.... Because of what the Midlands Examining Group described as "a procedural, rather than a computer error", none of the 12,000 entrants ... was awarded more than a grade C ...

.... Mr John Reddaway, secretary of the Cambridge board, said that because of misunderstandings between the five boards, the "hurdle" mark that distinguishes an A or B grade was not programmed into the computer. ...

[1 September 1986 - Letters to the Editor]

O-LEVEL ERRORS
>From Mr P.D.R. Talbot Willcox

Sir, The case reported in your columns today (August 28) of the computer error affecting the grades of O-level candidates raises the question whether other undetected computer errors are resulting in injustice and danger. The statement made by the Secretary of the Cambridge Board that "with new computer programmes [sic] we cannot find out mistakes until something happens" is hardly reassuring.

The error was sufficiently gross to excite determined questioning by those most obviously affected. But one dreads to think what might have happened if only a smaller number of pupils had been affected. There are many other computer applications where errors of this kind would have more serious and even disastrous implications, not least being medical and criminal records.

Is it not time for a Government enquiry to be held into ways and means of legislating to ensure that all potentially dangerous programmes are thoroughly checked before they are used?

Yours faithfully,
P.D.R. TALBOT WILLCOX, Rodwell House,Middlesex St, [London] E1, August 28.

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 49

Thursday, 4 September 1986

Contents

- Human Error
  - Dave Parnas
  - Bill Anderson
- Machine errors - another point of view
  - Bob Estell
- Flight simulators
  - Eugene Miya
- F-16 software
  - Henry Spencer
- Terminal (!) lockup
  - Ken Steiglitz
- Info on RISKS (comp.risks)

---

human error

Wed, 27 Aug 86 09:16:44 EDT

When people use the phrases "human error" and "computer error" they are simply trying to distinguish between situations in which "the cause" of the accident was a human action that happened about the time of the accident and the situations in which "the cause" of the error was a human action much earlier. Obviously, we cannot make a hard black/white distinction based on this continuum of possibilities. Only humans cause accidents because only humans provide the problem statements that allow one to talk about an accident or a failure.

Dave

Re: RISKS-3.46: Human Error

<WAnderson.wbst@Xerox.COM>
3 Sep 86 09:57 EDT

Herb Lin writes:

"no user should approach a computer system as though its behavior is predictable and/or sensible under all circumstances ...."

Stanislaw Lem has written some very amusing and thought provoking stories about the relations between people and technology (including automata) in the Tales of Pirx the Pilot (2 volumes, paperback). Pirx is just an ordinary space pilot who learns to approache the computer systems he must use with a good deal of common sense.

Bill Anderson

Machine errors - another point of view

<"SEFB::ESTELL" <estell%sefb.decnet@nwc-143b.ARPA<>
3 Sep 86 12:19:00 PST

I'm not satisfied with the notion that computers don't make errors; that they ONLY suffer mechanical failures that can be fixed.

"Deep in a computer's hardware are circuits called arbiters whose function is to select exactly one out of a set of binary singnals. If one of the signals can change from '0' to '1' while the selection is being made, the subsequent behavior of the computer may be unpredictable. It appears fundamentally impossible to construct an arbiter that can reliably make its selection within a bounded time interval."
Peter J. Denning, in AMERICAN SCIENTIST 73, no. 6 (Nov-Dec 1985)
[also reprinted in RIACS TR 85.12]

I'm not a hardware guru, nor a scholar in theoretical computer science; but my practical experience says that Peter is right. I've gotten very close to the internals of only two computers; both were IBM second generation machines, the 7074, and the 1401. I programmed both in assembly and machine code; even wrote diagnostics for the 7074. I can guarantee that those machines, much simpler in design than today's multi-processors, and also much slower, were somewhat unpredictable. We found some nasty situations that required special code loops to mask/unmask interrupts, so that the machine could run.

A "machine" as seen by the applications programmer, is already several layers [raw hardware, microcode, operating system kernel, run-time libraries, compiler]; and each layer is perhaps nearly a million pieces [IC's, lines of (micro)code] that may interact with nearly a million other pieces in other layers.

What I suspect here is a "problem of scale" akin to the well know idea that there are real limits to what one can build with a given material; e.g., bones can't support animals much over 100 feet tall; because the internal tensile and sheer stresses will at some point destroy the molecular integrity of the materials. We can analyse the few hundred lines of code, in the kernel of an I/O driver, running on naked second generation hardware; I did that. But can we examine the millions of lines of code that comprise the micro-instructions, the operating system, and the engineering applications on a multi-processor system, and hope to understand ALL the possible side-effects? Color me skeptical. Thus, because we put machines "in control" of significant events in our lives [ATM's, FFA stuff, weapons simulators, etc.]; and because EVEN AFTER we've made our best personal and professional attempt to eleminate the errors; and even after the system has run "a thousand test cases" it still has errors - not necessarily "hard failures" that the C.E. can fix, but "transients" that are sensitive to timing ; for all these reasons, I'll argue that "machines make errors" in much the same sense that people mispronounce words or make mistakes driving. It's not that we don't know better; it's not that we've suffered some damage; it's that we aren't perfect; neither are our computers. And sometimes there's "nothing wrong" that can be fixed.

If we continue this discussion long enough, we'll approach the metaphysical notion of "free will and determinism." I don't think that's necessary; I think our current systems have already exceeded our ability to predict them 100.0%, even in theory.

Bob

Flight simulators

Eugene Miya <eugene@AMES-NAS.ARPA>
3 Sep 1986 1714-PDT (Wednesday)

> [I thought that by now these simulators were designed so that they could be driven by the same software that is used in the live aircraft <PGN>...

Don't forget that very few aircraft use "software." Software is a radically new concept to aircraft designers: F-16, F-18, X-29A, and so forth.

> change in one place would be reflected by the same change in the other, although changing the application code without having to modify the simulator itself. Maybe not... PGN]

The problem comes when it's asked "What do you simulate?" The view? The feeling? Handling characteristic?->based on aerodynamics->computational fluid dynamics->??? True, those games your can buy for an apple two are simulators, and we have a $100 million test facility (6 stories high) which is a simulator. But there are limits to simulation:
we don't know how to simulate the flight characteristics of
a helicopter, we don't know how to automate a helicopter: (any one know of a microprocessor which can withstand 800-1600 Gs?). Anyway, Peter, you are invited to talk to our simulator people if you want to answer this one, as I don't have the time. Danny Cohen has been here.

Another thought: as simulators become more "real" [as in some of ours] they require increasing amounts of certification BEFORE you can use a simulator [does this sound like a paradox in some ways? hope so].

I saw an experienced pilot told they he could not use a simulator
in some mode (motion base).

--eugene miya, NASA Ames Research Center
{hplabs,hao,dual,ihnp4,decwrl,allegra,tektronix,menlo70}!ames!aurora!eugene

F-16 software

<allegra!utzoo!henry@ucbvax.Berkeley.EDU>
Wed, 3 Sep 86 23:59:56 PDT

Phil Ngai writes:

> It sounds very funny that the software would let you drop a bomb on the wing
> while in inverted flight but is it really important to prevent this? ...

This issue actually is even more complex than it sounds, because it may be *desirable* to permit this in certain circumstances. The question is not whether the plane is upside down at the time of bomb release, but which way the bomb's net acceleration vector is pointing. If the plane is in level flight upside-down, the vector points into the wing, which is a no-no. But the same thing can happen with the plane upright but pulling hard into a dive. Not common, but possible. On the other side of the coin, some "toss-bombing" techniques *demand* bomb release in unusual attitudes, because aircraft maneuvering is being used to throw the bomb into an unorthodox trajectory. Toss-bombing is common when it is desired to bomb from a distance (e.g. well-defended targets) or when the aircraft should be as far away from the explosion as possible (e.g. nuclear weapons). Low-altitude flight in rough terrain at high speeds can also involve quite violent maneuvering, possibly demanding bomb release in other than straight-and-level conditions.

Henry Spencer @ U of Toronto Zoology
{allegra,ihnp4,decvax,pyramid}!utzoo!henry

Terminal (!) lockup

<princeton!ken@seismo.CSS.GOV>
Wed, 3 Sep 86 01:34:17 EDT

From the User's manual for the Concept AVT terminal, p. 3-52 (Human Designed Systems, Inc., 3440 Market Street, Philadelphia, PA 19104):

"Note: since the Latent Expression is invoked automatically, it should not contain any command that resets the terminal, either implicitly or explicitly. If any such command is included in the Latent Expression, the terminal will go into an endless loop the next time it is reset (implicitly or explicitly) or powered up. The only way to break out of this loop is to disassemble the terminal and physically reset Non-Volatile Memory. ... "

Having a sequence of keystrokes that will physically disable a terminal seems to me a bad thing. For one thing it makes me awfully nervous when I'm changing the Latent Expression. For another, it opens up the possibility of having my terminal physically disabled by people or events outside my control. (I don't know whether this effect can also be caused by a sequence of bits sent to the terminal.)

I wonder: How common is this property of terminals (or other equipment)? Does the phenomenon have a (polite) name? Is it so hard to avoid that we should be satisfied to live with it? Is it clear how to test for the possibility? Does anybody have any experience with this phenomenon?

[You have found the tip of the iceberg of Trojan horses that can take over your terminal, processes, files, etc. PGN]

Ken Steiglitz, Dept. of Computer Science, Princeton Univ., Princeton, NJ 08544

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 50

Sunday, 7 September 1986

Contents

- Enlightened Traffic Management
  Alan Wexelblat
- Flight Simulator Simulators Have Faults
  Dave Benson
- Re: Flight Simulators and Software Bugs
  Bjorn Freeman-Benson
- Always Mount a Scratch Monkey
  Art Evans
- Re: supermarket crashes
  Jeffrey Mogul
- Machine errors - another point of view
  Bob Estell
- Human Behv. & FSM's
  Robert DiCamillo
- Info on RISKS (comp.risks)

Enlightened Traffic Management

Alan Wexelblat <wex@mcc.com>
Thu, 4 Sep 86 09:58:49 CDT

The Austin rag carried the following brief item off the AP wire:

NEW DELHI, India (AP) - The computer lost the battle with the commuter.

"Enlightened traffic management" was the term for New Delhi's new computerized bus routes, but four days of shattered windows, deflated tires and protest marches convinced the bus company that its computer was wrong.

The routes dictated by the computer proved exceedingly unpopular with passengers, who claimed that they were not being taken where they wanted to go.

Bowing to demand, the New Delhi Transport Corp. scrapped the new "rationalized" routes and restored 114 old routes.

"The computer has failed," shouted thousands of victorious commuters in eastern New Delhi Tuesday night after transport officials drove around in jeeps, using loudspeakers to announce the return of the old routes.

COMMENTS: At first, I thought this was pretty amusing; deflated tires is a computer risk I hadn't heard of before. But the whole attitude of the article (and seemingly the people) annoyed me. The machine is taking the rap and I'll bet that the idiot who programmed it to produce "optimal" routes will get off scott free. Not to mention the company execs who failed to understand their customer base and allowed the computer to "dictate" new routes. ARGH!

Alan Wexelblat
UUCP: {seismo, harvard, gatech, pyramid, &c.}!ut-sally!im4u!milano!wex

Flight Simulator Simulators Have Faults

Dave Benson <benson%wsu.csnet@CSNET-RELAY.ARPA>
Wed, 3 Sep 86 17:01:17 pdt

|I developed flight simulators for over 7 years and could describe many such
|bizarre incidents.

Might be interesting for RISKS if these suggest problems in developing risk-free software...

|To point out a failure during
|testing (or more likely development) seems meaningless. Failures that make
|it into the actual product are what should be of concern.

I do not agree. We need to understand that the more faults found at any stage to engineering software the less confidence one has in the final product. The more faults found, the higher the likelyhood that faults remain. I simply mentioned this one because it appears to demonstrate that for all the claims made for careful analysis and review of requirements and design, in fact the current practice leaves such obvious faults to be found by testing.

|As for the effectiveness of simulators...

Simulators are wonderful. Surely nothing I wrote suggested otherwise.

Upon further inquiry, the blank sky was in a piece of software used to simulate the flight simulator hardware. The software specs essentially duplicated the functions proposed for the hardware. So the hardware was going to take the trigonmetric tangent of the pitch angle. The software simulator of the flight simulator indeed demonstrated that one ought not take the tangent of 90 degrees.

So somebody with presumably a good background in engineering mathematics simply failed to think through the most immediate consequences of the trigonometric tangent function. Nobody noticed this in any kind of review, nobody THOUGHT about it at all.

Since nobody bothered to think, the fault was found by writing a computer program and then observing the obvious. I suggest that this inability to think bodes ill for the practice of software engineering and the introduction of "advanced techniques" such as fault-tree analysis.

I suggest that such examples of a pronounced inattention to well-known mathematics are part of the reason for such lengthy testing sequences as the military requires. And I suggest that the fact that it appears necessary to mention all this yet once again suggests that there are many people doing "software engineering" who have failed to grasp what a higher education is supposed to be about. I certainly do not expect perfection, but the trigonometric tangent is an example of an elementary function.

Re: Flight Simulators and Software Bugs

Bjorn Freeman-Benson <bnfb@uw-june.arpa>
Fri, 5 Sep 86 10:02:38 PDT

In RISKS-3.48, Gary Whisenhunt talks about how he developed flight simulators and that he "..seriously doubt[s] that the sky went blank in the B-1 simulator when it was delivered to the government." And then he goes on to point out all the specs it had to pass. I don't know no way or the other, but I want to point out that the sky going blank points out either a design problem or an implementation problem. If it is a design problem, who knows how many other serious (sky blanking serious) problems exist? Will the MIL standards catch them all? If it is an implementation error, who knows how many other similar coding errors that sloppy/tired/etc engineer made? If it's a sign problem, what happens when you back the plane up? Will it go into an infinite speed reverse? The point I'm trying to make is that bugs are not independent, and if one shows up, other similar are usually in existence.

Bjorn N Freeman-Benson
U of Washington, Comp Sci

Always Mount a Scratch Monkey

"Art Evans" <Evans@TL-20B.ARPA>
Wed 3 Sep 86 16:46:31-EDT

In another forum that I follow, one corespondent always adds the comment
Always Mount a Scratch Monkey
after his signature. In response to a request for explanation, he replied somewhat as follows. Since I'm reproducing without permission, I have disguised a few things.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

My friend Bud used to be the intercept man at a computer vendor for calls when an irate customer called. Seems one day Bud was sitting at his desk when the phone rang.
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Bud: Hello. Voice: YOU KILLED MABEL!!
B: Excuse me? V: YOU KILLED MABEL!!

This went on for a couple of minutes and Bud was getting nowhere, so he decided to alter his approach to the customer.

B: HOW DID I KILL MABEL? V: YOU PM'ED MY MACHINE!!

Well to avoid making a long story even longer, I will abbreviate what had happened. The customer was a Biologist at the University of Blah-de-blah, and he had one of our computers that controlled gas mixtures that Mabel (the monkey) breathed. Now Mabel was not your ordinary monkey. The University had spent years teaching Mabel to swim, and they were studying the effects that different gas mixtures had on her physiology. It turns out that the repair folks had just gotten a new Calibrated Power Supply (used to calibrate analog equipment), and at their first opportunity decided to calibrate the D/A converters in that computer. This changed some of the gas mixtures and poor Mabel was asphyxiated. Well Bud then called the branch manager for the repair folks:

Manager: Hello
B: This is Bud, I heard you did a PM at the University of
Blah-de-blah.
M: Yes, we really performed a complete PM. What can I do
for You?
B: Can You Swim?

The moral is, of course, that you should always mount a scratch monkey.

~~~~~~~~~~~~~~~~~~~~~~

There are several morals here related to risks in use of computers. Examples include, "If it ain't broken, don't fix it." However, the cautious philosophical approach implied by "always mount a scratch monkey" says a lot that we should keep in mind.

Art Evans
Tartan Labs

Re: supermarket crashes

Jeffrey Mogul <mogul@decwrl.DEC.COM>
4 Sep 1986 1614-PDT (Thursday)

One of the nearby Safeway supermarkets is open 24 hours, and is quite popular with late-night shoppers (it's known by some as the "Singles Safeway"). Smart shoppers, however, used to avoid visiting just before midnight, because that's when all the cash registers were out of operation while they went through some sort of ritual (daily balances or somesuch), simultaneously.
~~~~~~~~~~~~~~~~~~~~~~

I also discovered that this market, at least, is not immune to power failures; I was buying a quart of milk one evening when a brief blackout hit the area. The lights were restored within minutes, but the computer was dead and the cashiers "knew" it would be a long time before it would be up; they weren't about to waste their fortuitous coffee-break adding things up by hand, perhaps because they couldn't even tell the price of anything (or indeed, what it was, in the case of produce) without the computer.

I don't often shop at that market, partly because the markets I do use have cashiers who know what things are rather than relying on the computer. Some day, just for fun, I might mark a pound of pecans with the code number for walnuts, and see if I can save some money.

Machine errors - another point of view

<LENOIL@XX.LCS.MIT.EDU>
Thu, 4 Sep 1986 21:27 EDT

> A "machine" as seen by the applications programmer, is already several layers [raw hardware, microcode, operating system kernel, run-time libraries, compiler]; and each layer is perhaps nearly a million pieces [IC's, lines of (micro)code] that may interact with nearly a million other pieces in other layers.

Interaction between one million pieces of a system is more than just an exaggeration, it is horrendous engineering practice that should never be seen. Flow-graphs, dependency diagrams, top-down design - all are ways of reducing interaction between system components to a small, manageable size - the smaller the better. The probability of designing a working system of one million fully-connected components is near-zero. Furthermore, you seem to imply that component interconnects can transcend abstraction boundaries (e.g. microcode <-> run-time libraries); this again is poor engineering practice. I don't disagree that rising system complexity is a problem today, but you are several orders of magnitude off in your statement of the problem.

Robert Lenoil

Human Behv. & FSM's

Robert DiCamillo <rdicamil@cc2.bbn.com>
Fri, 5 Sep 86 16:27:45 EDT

Comments on Bob Estell's "Machine Errors", Risks Vol. 3, #49
(FSM's need friends too)

I have often felt the same way Bob Estell does - that the full scope of (software) engineering is too vast for a mere mortal to comprehend. However, I usually reassure myself with a good dose of computational theory:

* "... for all these reasons 'machines make errors' in much the same *

* sense that people mispronounce words or make mistakes driving." *

I agree with the apparent analogy, but still cringe at the actual usage of the word error. Webster's Ninth New Collegiate dictionary defines error as an "act involving unintentional deviation from truth or accuracy". If truth or accuracy for computers or finite state automata is defined to be the mapping of all possible input states to output states, then theoretically, the only *unintentional* deviation from such truth (tables or such) is the failure to map or correlate all possible input strings to known or desired output states.

I have participated in the situation where the adoption of a non-standard arbitration scheme did not take into account cycle stealing, and assembly code actually had the value of operands corrupted so that a branch occurred on the opposite condition to the true data. This was a bug that only a logic analyzer could find, and set the hardware engineers back to their drawing board. You have no idea how strange it feels to tell someone, that the code actually took a branch wrong; prior to the branch the data was true, but it always branched to the false address. The high level DDT would never show the data to be false because of the particular timing coincidences involved with using an in circuit emulator; very disturbing when even your debugger says all is well, and tests still fail operationally in the real system.

In the case of bus arbitration, an entire realm of undesirable input strings should be eliminated if the timing constraints between competing processes are properly enforced in hardware. If they are not, "unintentional deviation" from the arbitration scheme will occur, but that "deviation" is really only another set of output states that serves no desirable function. However, you could sit down with a logic analyzer and painfully construct a mapping of all possible input timing states to a bus arbitration scheme, and map the output. Hopefully, the design engineers did this when they made the specifications, even if they were not exhaustive in testing every possible input string.

I believe it is improper to construe human behavior, especially *unpredictability* to the results of input strings that fall outside the desired function of a finite state automata. In theory, a FSM can have an undefined output for a given input, but in practice the definition of this output usually depends upon the resolution of your measuring instruments. If an arbitration scheme appears to yield an indeterminate output when all inputs are still within spec (proper input strings), then the characteristic function of the FSM is not complete (well defined). Practically, this could mean that a timing situation arose they couldn't or didn't see - maybe their analyzer didn't have the resolution ? But it is still ultimately, and sometimes easily attributable to a human oversight. How much of the FSM characteristic function do you know about ? The part you never dealt with is not necessarily "unpredictable".

Many important computational theories hinge on the conception that any "solvable" problem can be realized in an arbitrarily complex FSM. While it may not be practical to build the machine, no one yet has been able to disprove such assertions as Church's thesis with current silicon built architectures. Computational theory still clings to this viewpoint, which I practically see as - if output states seem indeterminate, you still haven't found the correct way to cast inputs in a reliably measurable form.

* "But can we examine the millions of lines of code that comprise the *
* micro-instructions, the operating system, and the engineering *
* applications on a multi-processor system, and hope to understand *
* ALL the possible side-effects." *

Goals of good software/hardware design are to make it easy to categorize all possible input strings, especially when they are countably infinite. This is not the same as viewing the machine as somehow irrational and unpredictable. Good designs may have an ease to their completeness of their characteristic function (CF). This does not mean bad designs are unpredictable, just maybe too complex to realize or measure. Anthropomorhizing is all too tempting. Systems with many architectural layers have complex interactions. Recent discussion in RISKS has highlighted the small percentage of total execution paths that are ever actually traced, but perhaps in well characterized FSM's, such exhaustive testing can be cautiously minimized. If in fact the range of the CF is countably infinite, then some method of limited testing is usually mandatory. Its the part of the FSM you don't know that you tend to ascribe human behavior to !

Maybe it does take some exposure to developing systems with both complete and incomplete characteristic functions to get an intuition about how closed the FSM has to be to give satisfactory performance, for a specific application. Bus arbitration is a relatively critical control function in most architectures, and should be given a high priority. I'm sure there are many systems out there that work just on the verge of catastrophe as sloppily implemented FSM's, at numerous levels.

Writing microcode, I tend to look at design issues architecturally; however, some experts believe that new architectures may be invented that will not be encompassed by contemporary computational theory. In the August 1986 SPECTRUM (from IEEE), the series of articles on optical computing addresses this problem:

* "In C. Lee Giles view, (program manager of the Air Force Office of *
* Scientific Research in Washington, D.C.), theoretical computer *
* science has 'stuck its neck out' by saying that computational *
* models define anything that is computable, since it is unknown *
* whether there are tasks these models cannot perform that the *
* human brain can." ..... *
* *
* (from the author Trudy E. Bell), "it remains to be seen whether *
* (optical) neural network architectures represent a new *
* computational model."

I would love to prove some philosophers wrong about how "computable tasks" can ultimately be cast in the form of FSM's. The dawn of the general purpose optical computer architecture may well introduce new models that require a new breed of non FSM computational theory. However, I think that computer engineering will focus on getting good "old fashioned" FSM's to work in the real world for a long time, and even at this level of complexity there will always be bugs from human behavior, not "machine behavior".

Search RISKS using swish-e

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 51

Sunday, 7 September 1986

Contents

- Computer almost created swing vote
  Bjorn Freeman-Benson
- Computer Sabotage of Encyclopedia Brittania
  Rosanna Lee
- F-16 software
  Wayne Throop
- Arbiter failures and design failures
  Martin Harriman
- Systems errors (hardware AND humans)
  Bill Janssen
- Re: Terminal (!) lockup
  Roy Smith
- Info on RISKS (comp.risks)

Computer almost created swing vote

Bjorn Freeman-Benson <bnfb@uw-june.arpa>
Sun, 7 Sep 86 10:44:01 PDT

Quoted without permission from the Seattle Times, Sunday Sep. 7, 1986:

AP, PHOENIX, Ariz. -- Tuesday's primary elections in Maricopa County would have been a mess if officials hadn't figured out that a compuer was set up to give all Republican votes to the Democrats and vice versa.

"If it had gone undetected, there would have a major, major problem with the election," County Recorder Keith Poletis said Friday. Poletis said that, if the compuer hadn't been fixed, a race with three Republicans and one Democrat would given the Democrat's votes to one of the Rebulicans.

Votes cast for the remaining Republicans would have been zapped into the void by the computer, because the software would find no Democratic opponents. "The computer sorts the cards into two piles, and it was sorting the Republicans into the Democrats' slots and the Democrats into the Republican slots," Poletis said.

A clerical error made the computer's cards were ordered was to blame for the mix-up, said Joe Martina, director of the county's computer systems. The error was caught during the secretary of state's test of the country's cards late Thursday.

End quote.

In my mind, this brings up an interesting question: should errors like this be reported (1) to the general public and (2) to the software engineering community? I think the answer to (2) is yes -- the more data we have on the types of errors that occur involving computers, the better grasp we will have on solving them. However, for (1), I see this arguement:

Con - The testing procedures before acceptance caught the error.
- The public will just lose faith in computers.
Pro - The public should know, because what if the testing hadn't?
Con - The public in general is not knowledgeable about computers and software, and the general press is sensationalist. Thus any case reported will not be studied in the necessary depth.
- An analogy with civil engineering: should the public know that the first design for a bridge collapsed during testing? Or is it just enough to know that the final bridge works?

Bjorn N Freeman-Benson

Computer Sabotage of Encyclopedia Brittania

Rosanna Lee <rosanna@CSL.SRI.COM>
Sat 6 Sep 86 18:09:51-PDT

Chicago Tribune [From San Jose Mercury News, Friday, Sept 5, 1986]

LAID-OFF WORKER SABOTAGES ENCYCLOPEDIA

CHICAGO - An employee of the Encyclopedia Britannica, disgruntled at having been laid off, apparently committed computer sabotage by altering portions of the text being prepared for updated editions of the renowned encyclopedia.

The discovery has prompted an exhaustive check to ensure that painstaking work, in the words of the editor, "is not turned into garbage."

"We have uncovered evidence of deliberate sabotage in the EB computer files," editor-in-chief Tom Goetz disclosed in an Aug. 28 memo to editorial personnel at the chicago headquarters of the oldest continually published reference work in the English language.

The unidentified former employee has confessed and is helping undo the damage, a spokesman said, although the company may press criminal charges. He said the 44-million word 1987 edition is safe, but employees are believed to be laboring overtime to catch alterations that could find their way into the 1988 edition.

Among the former employee's more vivid changes, sources said, was changing references to Jesus Christ to Allah, the Moslem name for God.

Goetz declined to comment Thursday other than to say, "Everything is under control." Another industry executive said, "In the computer age, this is exactly what we have nightmares about."

In the first of three memos to editorial staffers, Goetz wrote, "What is perhaps most distressing for each of us is the knowledge that some of our hard work has been turned into garbage by either a very sick or a very vicious person."

At the time, he said that the actions constituted a crime under Illinois law, that the company planned to pursue legal actions "vigorously" and that it was issuing new computer passwords to employees.

In a staff memo dated Wednesday, Goetz informed employees that "we have successfully concluded the matter of the sabotage of the encyclopedia's data base.

"The 1987 printing is secure," Goetz stated.

The publication first was alerted to a problem, sources said, when a worker scanned the computer data base and discovered the clearly odd insertion of the names of a company executive and a private consulting firm apparently

[There are several problems in believing that this audit-trail approach is fool-proof. First of all, it relies on a password. Masquerading is therefore a concern. The second is probably more important -- any self-respecting system programmer or cracker is probably able to alter the audit trail. It is dangerous to assume that the only disgruntled employess are those who are NOT computer sophisticates... PGN]

F-16 software

<rti-sel!dg_rtp!throopw%mcnc.csnet@CSNET-RELAY.ARPA>
Fri, 5 Sep 86 13:19:25 edt

> It sounds very funny that the software would let you drop a bomb on the wing
> while in inverted flight but is it really important to prevent this? Is it
> worth the chance of introducing a new bug to fix this very minor problem?

> [The probability is clearly NONZERO. It is very dangerous to start
> making assumptions in programming about being able to leave out an
> exception condition simply because you think it cannot arise. Such
> assumptions have a nasty habit of interacting with other assumptions
> or propagating. PGN]

It is also dangerous to start making assumptions about the ways in which the system will be used. Can you really not think of a reason why one would want to "drop" a bomb while the dorsal surface of the plane points towards the planet's center (a possible interpretation of "inverted")? I can think of several.

I am trying to make the point that the gross simplification of "preventing bomb release while inverted" doesn't map very well to what I assume the actual goal is: "preventing weapons discharge from damaging the aircraft". This is yet another instance where the assumptions made to simplify a real-world situation to manageable size can easily lead to design "errors", and is an architypical "computer risk" in the use of relatively simple computer models of reality.

In addition to all this, it may well be that one doesn't *want* to prevent all possible modes weapons discharge that may damage the aircraft... some of them may be useful techniques for use in extreme situations.

  The more control,
  The more that requires control.
  This is the road to chaos.
                      --- PanSpechi aphorism {Frank Herbert}

Wayne Throop   <the-known-world>!mcnc!rti-sel!dg_rtp!throopw

Arbiter failures and design failures

Martin Harriman <MARTIN%SRUCAD%sc.intel.com@CSNET-RELAY.ARPA>
Fri, 5 Sep 86 09:38 PDT

Bob Estell raises two quite different failure mechanisms in his message. The first mechanism he mentions is the well known problem of making a reliable arbiter; he then goes on to discuss the quite different problem of design errors in hardware, microcode, or systems software.

The arbiter problem is well known; fundamentally, there is no absolutely reliable way to sample asynchronous signals in a synchronous system, though there are ways of greatly reducing the probability of failure. In this sense, no computer which incorporates asynchronous interrupts is deterministic, since you can not predict its behavior cycle by cycle. It is important to take this effect into account in the design of the system and any software which cares about the timing of these external events.

There are other interesting failure modes, where the arbiter essentially says "maybe," instead of giving a clear yes or no answer; careful circuit design can reduce the probability of these failures to one failure every few thousand years (at least according to our last set of simulations).

The bulk of Bob's message is a discussion of the probability of design bugs. Anyone who has seen the errata sheets for a microprocessor or the ECO history of a mainframe will know that computer hardware is imperfect. This may be news to some computer programmers, but there is such a thing as a computer error; for instance, the first stepping of Intel's 80186 was convinced that the product of two negative numbers was negative.

--Martin Harriman (martin%srucad@sc.intel.com)
Intel Santa Cruz

Systems errors (hardware AND humans)

Bill Janssen <janssen@mcc.com>
Fri, 5 Sep 86 18:07:08 CDT

Bob Estell's note on machine errors made me think of an error that I found some years ago. I was writing a C program that, among other things, provided a virtual connection between two serial ports. The code looked something like this:

```
  while (in_connection_mode)
     {
     if (input_available(port1))
        port2->output = port1->input;
     else if (input_available(port2))
        port1->output = port2->input;
     }
```

where `port1' and `port2' were pointers to register banks on the serial port controllers. When we tried it out, it didn't work. To make a long story short, it didn't work because assembly code for "port2->output = port1->input" was produced very efficiently as (something like) "MOVB 4(A4),8(A5)", which would still have been OK, except that both serial ports were on the same chip and the chip needed a recovery interval after doing a read before doing a write. Working code used the line "port2->output = temp = port1->input", to introduce a slight delay!!

Now, where's the source of the error here? What bugs me is that you can PROVE that the (non-functional) code functions properly... if you ignore the hardware quirks, which aren't documented. And if the compiler produced less efficient code (load register; store register) the HLL code would work. And if the machine architecture didn't have memory-to-memory move instructions the code would work. And if the computer clock was slower, the code would work. I tend to think that the error was in the characterization of the hardware, which described the two serial ports as independent. But perhaps the error is actually in not VERIFYING the hardware characterization...

Bill
--
Bill Janssen, MCC Software Technology
9430 Research Blvd, Austin, Texas 78759
ARPA: janssen@mcc.com PHONE: (512) 339-3682
UUCP: {ihnp4,seismo,harvard,gatech,pyramid}!ut-sally!im4u!milano!janssen

---

Re: Terminal (!) lockup

Roy Smith <cmcl2!phri!roy@seismo.CSS.GOV>
Fri, 5 Sep 86 18:23:34 edt

> I wonder: How common is this property [being able to break it
> by pushing the wrong combination of buttons] of terminals (or
> other equipment)?

We have some CTS-2400 auto-dial modems that let you set all sorts of parameters that get stored in eeprom. It's not too hard to set it up so it doesn't echo and doesn't produce any output at all. This condition persists even after power-cycling. It's not really dead, but unless you realize what you did and know the magic sequence to turn back on echoing and command processing, it sure looks that way (if it looks like a duck and quacks like a duck ...)

Take a typical time-sharing system, erase the boot block from disk and turn it off. You've sure done a nice imitation of breaking it (I consider having to toggle in a binary boot program as very much akin to opening up a terminal to fix it). If you've got a writeable control store, you could mess yourself up even more.

The (clever) people who designed the Apple LaserWriter must have been thinking along these lines. There are 2 serial interfaces on the LW. You can run a little PostScript to change the baud rate (stored in eeprom) on either or both. If you want to disable one interface, you just set its baud rate to 0. According to the documention (I've never tried it :-)) it won't let you set both channels to 0 baud. If you could, there would be no way to talk to it short of yanking the eeprom.

Search RISKS using swish-e

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 52

Tuesday, 9 September 1986

Contents

- Re: F-16 software
  Nancy Leveson
- Upside-down F-16's and "Human error"
  Jon Jacky
- F-16 software
  Scott E. Preece
- Do More Faults Mean More Faults?
  Ken Dymond
- Why components DON'T interact more often
  Bob Estell
- Computer almost created swing vote
  Scott E. Preece
- Computer Sabotage [MISSING LAST LINE FROM RISKS-3.51]
- Computer Sabotage of Encyclopedia Brittanica
  Scott E. Preece
- Captain Midnight & military satellites
  Werner Uhrig
- Re: always mount a scratch monkey
  Alexander Dupuy
- Erroneous computer printout used in public debates
  Chris Koenigsberg
- Info on RISKS (comp.risks)

Re: F-16 software

Nancy Leveson <nancy@ICSD.UCI.EDU>
08 Sep 86 09:53:29 PDT (Mon)

Wayne Throop writes:

>it may well be that one doesn't *want* to prevent all possible
>modes weapons discharge that may damage the aircraft ... some of
>them may be useful techniques for use in extreme situations.

This raises some extremely important points that should be remembered by those attempting to deal with risk.

1) nothing can be made 100% safe under all circumstances. In papers I have written I have pointed out that safety razors and safety matches are not completely safe, they are only *safer* than their alternatives. Drinking water is usually considered safe, but drinking too much water can cause kidney failure.

1) the techniques used to make things safer usually involve limiting functionality or design freedom and thus involve tradeoffs with other desirable characteristics of the product.

All we can do is attempt to provide "acceptable risk." What is "acceptable" will depend upon moral, political, and practical issues such as how much we are willing to "pay" for a particular level of safety.

I define "software safety" as involving procedures to ensure that the software will execute within a system context without resulting in unacceptable risk. This implies that when building safety-critical systems, one of the first and most important design problems may be in identifying the risks and determining what will be considered acceptable risk for that system. And just as important, our models and techniques are going to have to consider the tradeoffs implicit in any attempt to enhance safety and to allow estimation of the risk implicit in any design decisions.
If we have such models, then we can use them for decision making, including the decision about whether acceptable risk can be achieved (and thus whether the system can and should be built). If it is determined that acceptable risk can be achieved, then the models and techniques should provide help in making the necessary design decisions and tradeoffs.
The important point is that these decisions should be carefully considered and not subject to the whim of one programmer who decides in an ad hoc fashion whether or not to put in the necessary checks and interlocks.

Nancy Leveson
Information & Computer Science
University of California, Irvine

Upside-down F-16's and "Human error"

Jon Jacky <jon@uw-june.arpa>
Mon, 8 Sep 86 16:55:19 PDT

> (... earlier postings mentioned "fly-by-wire" F-16 computer would
> attempt to raise landing gear while aircraft was sitting on runway,
> would attempt to drop bombs while flying inverted, and other such
> maneuvers -- in response to pilot's commands

These are regarded as errors? Maybe I'm missing something, but it sounds like the right solution is to remind the pilots not to attempt obviously destructive maneuvers. I detect a notion floating about that software should prevent any unreasonable behavior. This way lies madness. Do we have to include code to prevent the speed from exceeding 55 mph while taxiing down an interstate highway?

My point is, if you take the approach that the computer is supposed to check for and prevent any incorrect behavior, then you have saddled yourself with the task enumerating every possible thing the system should NOT do. Such a list of prohibited behaviors is likely to be so long it will make the programming task quite intractable, not to mention that you will never get all of them.

I suggest that the correct solution is the time-honored one: the operator must be assumed to possess some level of competence; no attempt is made to protect against every conceivable error that might be committed by a flagrantly incompetent or malicious operator.

Note that all non-computerized equipment is designed this way. If I steer my car into a freeway abutment, I am likely to get killed. Is this a "design flaw" or an "implementation bug?" Obviously, it is neither. People who are drunk or suicidal are advised not to drive.

This relates to the ongoing discusssion about "human error." This much-abused term used to refer to violations of commonly accepted standards of operator performance -- disobeying clear instructions, attempting to work when drunk, things like that. Apparently it has come to refer to almost any behavior which, in retrospect, turns out to have unfortunate consequences. It is sometimes applied to situations for which the operator was never trained, and which the people who installed the system had not even anticipated.

When abused in this way, the term "human error" can be a transparent attempt to deflect blame from designers and management to those with the least control over events. Other times, however, it is evidence of genuine confusion over who is responsible for what. Right at the beginning, designers must draw a clear line between what the automated system is supposed to do and what the operators must do. This may require facing the painful truth that there may be situations where, if the operator makes a mistake, a real disaster may occur. The choice is then one of ensuring the trustworthiness of the operators, or finding an alternative approach to the problem that is more robust.

I suggest that if additional computer-based checking against operator errors keeps getting added on after the system has been installed, it is evidence that the role of the operator was not very clearly defined to begin with.

-Jonathan Jacky
University of Washington

F-16 software

"Scott E. Preece" <preece%ccvaxa@GSWD-VMS.ARPA>
Mon, 8 Sep 86 09:36:42 cdt

> From: amdcad!phil@decwrl.DEC.COM (Phil Ngai)

> It sounds very funny that the software would let you drop a bomb on the
> wing while in inverted flight but is it really important to prevent
> this?

Others have already pointed out that sometimes you may WANT to release the bomb when inverted. I would ask the more obvious question: Would a mechanical bomb release keep you from releasing the bomb when inverted? I tend to doubt it. While it's nice to think that a software controlled plane should be smarter than a mechanical plane, I don't think it's fair to cite as an error in the control software that it isn't smarter than a mechanical plane...

If, in fact, the mechanical release HAD protected against inverted release, I would have expected that to be part of the specs for the plane; I would also expect that the acceptance tests for the software comtrolled plane would test all of the specs and that the fault would have been caught in that case.

scott preece
gould/csd - urbana
uucp: ihnp4!uiucdcs!ccvaxa!preece
arpa: preece@gswd-vms

Do More Faults Mean More Faults?

"DYMOND, KEN" <dymond@nbs-vms.ARPA>
8 Sep 86 09:18:00 EDT

In RISKS 3.50 Dave Benson comments in "Flight Simulator Simulators Have Faults" that

>We need to understand that the more faults found at
>any stage to engineering software the less confidence one has in the
>final product. The more faults found, the higher the likelyhood that
>faults remain.

This statement makes intuitive sense, but does anyone know of any data to support this ? Is this true of any models of software failures ? Is this true of the products in any of the hard engineering fields -- civil, mechanical, naval, etc. -- and do those fields have the confirming data ?

Ken Dymond, NBS

---

why components DON'T interact more often

"SEFB::ESTELL" <estell%sefb.decnet@nwc-143b.ARPA>
8 Sep 86 08:12:00 PST

I guess I neglected to emphasize a key word: "MAY."

My original posting said "...may interact..."
I am well aware that components SHOULD *NOT* interact.
I am also well aware that hardware designers labor to make sure that
the actual interactions are
 (1) very infrequent; and
 (2) not terribly damaging when they inevitably do occur.
Similarly, software designers [good ones!] labor to restrict the
inevitable interactions; and limit the damage done when they occur.
Since each layer of a well designed, carefully implement system
"filters" faithfully, the result is a system that will run for months
[years?] without random failures.
But random failures do occur. My ancient stories were replaced by more
current ones in later RISKS postings.
Until the theory and the practice of computing systems design EACH admit
that random error [including, but not limited to, interactions] is real,
we'll continue to build systems and applications less reliable than they
could be - or should be.

Bob

Computer almost created swing vote

"Scott E. Preece" <preece%ccvaxa@GSWD-VMS.ARPA>
Mon, 8 Sep 86 10:02:18 cdt

> From: bnfb@uw-june.arpa (Bjorn Freeman-Benson)
> In my mind, this brings up an interesting question: should errors like
> this be reported (1) to the general public and (2) to the software
> engineering community?

I don't think errors like this should be HIDDEN, but I also don't
think this demands issuing a press release. The reason you do
a test is to determine whether your procedures are working --
it shouldn't be thought newsworthy that you find mistakes in
testing. If, on dry run day, a manual election counting system
had mistakenly recorded the Democratic votes on the master tally
sheet on the Republican line and vice versa, the counter would
have been apprised of the error and instructed in proper
procedure, but I don't think they'd have issued a press release.

The problem in this particular case wasn't that the system
didn't work, but that the operators didn't understand the
operating procedures. That's no big deal, but the election
judges should be warned what to look for on election night to
see that the control information is correctly set up (regression
testing).

--
scott preece
gould/csd - urbana
uucp: ihnp4!uiucdcs!ccvaxa!preece
arpa: preece@gswd-vms

Computer Sabotage [LAST LINE MISSING FROM RISKS-3.51]

Rosanna Lee <rosanna@CSL.SRI.COM>
Sat 6 Sep 86 18:09:51-PDT

[LAST LINE INADVERTENTLY TRUNCATED... COMPLETE LAST PARAGRAPH FOLLOWS.]
The publication first was alerted to a problem, sources said, when a worker scanned the computer data base and discovered the clearly odd insertion of the names of a company executive and a private consulting firm apparently viewed by the former employee as partly responsible for the layoff decision.

Computer Sabotage of Encyclopedia Brittanica

"Scott E. Preece" <preece%ccvaxa@GSWD-VMS.ARPA>
Mon, 8 Sep 86 10:12:54 cdt

> [There are several problems in believing that this audit-trail approach
> is fool-proof. First of all, it relies on a password. Masquerading is
> therefore a concern. The second is probably more important -- any
> self-respecting system programmer or cracker is probably able to alter
> the audit trail. It is dangerous to assume that the only disgruntled
> employess are those who are NOT computer sophisticates... PGN]

Clearly the audit trail is not enough to protect against insider damage by systems programmers, but the article says nothing about whether there are other tools designed to deal with such users -- just that audit trail methods were sufficient in this case. Let's not jump to conclusions.

Curiously, embezzlement, fraud, doctored documents, and disgruntled employee sabotage all pre-dated computers. It appears to be the case (I haven't heard the details yet) that in this case the fact that the system was computerized allowed them to identify the damage quickly and repair it. If the files were on paper and the saboteur had simply altered and replaced random pages of random articles in the files, the damage would have been worse and much harder to trace and fix. The system doesn't have to be foolproof to be an improvement over manual systems.

--
scott preece
gould/csd - urbana
uucp: ihnp4!uiucdcs!ccvaxa!preece
arpa: preece@gswd-vms

Captain Midnight & military satellites (Mother Jones, October 86)

Werner Uhrig <CMP.WERNER@R20.UTEXAS.EDU>

Mon 8 Sep 86 00:01:30-CDT

[pointer to article in print: Mother Jones, Oct '86 Cover Story on Satellite
Communications Security (or lack thereof)]

(p.26) CAPTAIN MIDNIGHT, HBO, AND WORLD WAR III - by Donald Goldberg
John "Captain Mignight" MacDougall has been caught but the flaws he exposed in the U.S. military and commercial ssatellite communications system are still with us and could lead to far scarier things than a $12.95 monthly cable charge.

(p.49) HOME JAMMING: A DO-IT-YOURSELF GUIDE - by Donald Goldberg
What cable companies and the Pentagon don;t want you to know.

PS: Donald Goldberg is described as "senior reporter in Washington, D.C., for the syndicated Jack Anderson column

[this is not an endorsement of the article, just a pointer.
you be the judge of the contents.]

Re: always mount a scratch monkey

Alexander Dupuy <dupuy%amsterdam@columbia.edu>
Mon, 8 Sep 86 03:00:30 EDT

Here's another version of this story, from the ever reliable usenet net.rumor. The existence of the alternate versions puts both pretty much in the realm of apocrypha. It's still a good story though...

From: moroney@jon.DEC (Mike Moroney)
Newsgroups: net.rumor
Subject: Re: Computer war stories
Date: 19 Mar 86 18:19:22 GMT
Organization: Digital Equipment Corporation

Yet another old classic war story.
--

It seems that there was a certain university that was doing experiments in behavior modification in response to brain stimulation in primates. They had this monkey with a number of electrodes embedded in it's brain that were hooked up to a PDP-11. They had several programs that would stimulate different parts of the monkey's brain, and they had spent over a year training the monkey to respond to certain stimuli. Well, eventually the PDP developed problems, and field service was called in. Due to some miscommunication, the field service representative was not informed of the delicacy of this particular setup, and the people running the experiment were not informed that field service was coming to fix the machine. The FS representative then booted up a diagnostic system I/O exerciser. After several minutes of gyrations, the monkey expired, it's brain fried.

The moral, of course, is "Always mount a scratch monkey"

Erroneous computer printout used in public debates

Chris Koenigsberg <ckk@andrew.cmu.edu>
Mon, 8 Sep 86 10:23:56 edt

[Brief background on this story: In Pennsylvania, all sales of wine and hard liquor are made at State Stores, run by the Liquor Control Board. The Governor has been trying to abolish the board and let private industry take over the liquor business, but LCB employee unions have been fighting against him. The LCB held a 20% discount sale on the Saturday before Labor Day, and the unions were outraged because the LCB's mission is actually to control alcohol, not promote it, and the sale seemed to encourage consumption on the holiday weekend. The debate over how much was sold, how much profit or loss was made, and the effects on holiday weekend drunk driving were hot news all week. Now this report of a computer error comes after public debate already occurred, in which people relied on the incorrect sales figures.]

++

Article from the Pittsburgh Post-Gazette, Saturday, September 6, 1986, written by Gary Rotstein (copyright 1986 PG Publishing Co.)

"20% discount sale brought LCB $18.9 million, not $8.5 million"

Admitting a $10 million flub in a computer printout, the Pennsylvania Liquor Control Board reported yesterday that it sold a one-day record high of $18.9 million of alcohol last Saturday. LCB Chairman Daniel Pennick told reporters Wednesday that the second 20 percent discount sale in the agency's history had grossed only about $8.5 million. That would have been $5 million more than was sold on the comparable date a year ago, but less than the $11 million one-day high recorded during a similar sale in June.

LCB spokesman Robert Ford said the agency's comptroller's office reviewed the figures yesterday morning and realized an important digit - the numeral 1 indicating $10 million - had been unable to fit on the initial computer printout tallying the sales figure. Once a correction was made and final purchases from Saturday were tacked on, the LCB learned it had sold $18.9 million in goods.

Ford noted that the comptroller's office personnel responsible for the mistake are employees of the governor's budget office rather than the LCB.

"The fact that someone made an error doesn't bother us," Ford said. "We're just happy about the sales figures."

Whether the higher sales is good or bad news for the LCB, however, is in dispute between the agency and its longtime critic, Governor Thornburgh. Thornburgh's budget office has estimated, based on an analysis of the LCB's receipts and costs last year, that when the price of a bottle is reduced by 20 percent the agency loses an average of $1.13 on each item sold. That scenario means it's worse for the LCB's financial picture to have $18.9 million in discount sales than $8.5 million, administration spokesman Michael

Moyle pointed out.

Ford maintained that the sale only cut into the size of the LCB's profits and did not actually amount to a net loss.

"We didn't lose a penny on any bottle sold," he said.

++

[notice that Ford emphasizes how the mistake was made by the governor's budget office (the same office responsible for the disputed estimate that a 20% sale would lose $1.13 on each item), and not by his LCB employees - his agency is under fire and is close to being dissolved by the legislature. But it's made very clear that a human error led to the missing digits, rather than trying to "blame it on the computer"]

Christopher Koenigsberg
Center for Design of Educational Computing
Carnegie-Mellon University
(412)268-8526
ckk@andrew.cmu.edu

Search RISKS using swish-e

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 53

Wednesday, 10 September 1986

Contents

- Hardware/software interface and risks
  Mike Brown
- More on Upside down F-16s
  Mike Brown
- "Unreasonable behavior" and software
  Gary Chapman
- Re: supermarket crashes
  Scott Preece
- Info on RISKS (comp.risks)

Hardware/software interface and risks

<mlbrown@nswc-wo.ARPA>
Tue, 9 Sep 86 09:48:31 edt

In RISKS 3.51 Bill Janssen writes of errors made in failing to consider the interaction of the hardware and the software under design. This failing was all too common in the writing of assembly and machine code during the early days of programming. Discrete wired machines often had OP codes that were not generally well known (i.e. the computer designers kept it secret). Interestingly, these unknown OP codes were included when more modern machines emulated the original discrete design. An excellent example is the old IBM 4-PI CP-3 and the IBM 360.

The hardware/software interface within the machine can create significant problems from both a software safety and software security standpoint. Software designers will have to have an increasingly detailed knowledge of the total system to produce the safe, secure software that critical systems will require.

More on Upside down F-16s

<mlbrown@nswc-wo.ARPA>
Tue, 9 Sep 86 10:00:00 edt

In RISKS 3.52 Jon Jacky writes:

>..it sounds like the right solution is to remind the pilots not to attempt
> obviously destructive maneuvers. ...if you take the approach that the
> computer is supposed to check for and prevent any incorrect behavior, then
> you have saddled yourself with the task enumerating every possible thing the
> system should not do."

Perhaps a solution is to remind the pilots not to attempt obviously destructive maneuvers however, relying on procedures to eliminate or reduce the risk of hazards is the least acceptable way. Pilots are human and as such are prone to making errors. Look at the safety record for general aviation and the Navy - both are dismal and are often reported to be due to pilot error. Its fine to tell the pilot "Lower your wheels before you land, not after" but we still have gear up landings. We should not concern ourselves with checking for and preventing any incorrect behavior but we should preclude that behavior which will result in damage to or loss of the aircraft or the pilot. We do not need to anticipate every possible mistake that he can make in this regard either - all we need to do are to identify the hazardous operational modes and prevent their occurrence.

Mike Brown, Chairman Triservice Software Systems Safety Working Group

"Unreasonable behavior" and software

Gary Chapman <chapman@russell.stanford.edu>
Tue, 9 Sep 86 14:28:24 pdt

Jon Jacky wrote:

> I detect a notion floating about that software should prevent any unreasonable behavior. This way lies madness. Do we have to include code to prevent the speed [of an F-16] from exceeding 55 mph while taxiing down an interstate highway?

I certainly agree with the thrust of this. But we should note that there is plenty of evidence that coding in prohibitions on unreasonable behavior will be required, particularly in the development of "autonomous" weapons that are meant to combat the enemy without human "operators" on the scene.

Here's a description of a contract let by the United States Army Training and Doctrine Command (TRADOC), Field Artillery Division, for something called a "Terminal Homing Munition" (THM):

> Information about targets can be placed into the munitions processor prior to firing along with updates on meteorological conditions and terrain. Warhead functioning can also be selected as variable options will be available. The intro-

duction of VHSIC processors will give the terminal homing
munitions the capability of distinguishing between enemy and
friendly systems and finite target type selection. Since
the decision of which target to attack is made on board the
weapon, the THM will approach human intelligence in this area.
The design criteria is pointed toward one munition per target
kill.

(I scratched my head along with the rest of you when I saw this; I've always thought if you fire a bullet or a shell out of a tube it goes until it hits something, preferably something you're aiming at. But maybe the Army has some new theories of ballistics we don't know about yet.)

As Nancy Leveson notes, we make tradeoffs in design and functionality for safety, and how many and what kinds of tradeoffs are made depends on ethical, political and cost considerations, among other things. Since, as Jon Jacky notes, trying to prohibit all unreasonable situations in code is itself unreasonable, then one wonders what sorts of things will be left out of the code of terminal homing munitions? What sorts of things will we have to take into account in the code of a "warhead" that is supposed to find its own targets? What level of confidence would we have to give soldiers (human soldiers--we may have to get used to using that caveat) operating at close proximity to THMs that the things are "safe"?

I was once a participant in an artillery briefing by a young, smart artillery corps major. This officer told us (a bunch of grunts) that we no longer needed "forward observers," or guys attached to patrols to call in the ranges on artillery strikes. In fact, said the major, we don't need to call in our artillery stikes at all--his methods had become so advanced would just know where and when we needed support. We all looked at him like he had gone stark raving mad. An old grizzled master sergeant who had been in the Army since Valley Forge I think, got up and said, "Sir, with all due respect, if I find out you're in charge of the artillery in my sector, I will personally come back and shoot you right between the eyes." (His own form of THM "approaching human intelligence", no doubt.) (I wouldn't be surprised if this major wrote the language above.)

What is "unreasonable" behavior to take into account in coding software? The major's or the sergeant's?

-- Gary Chapman

Re: supermarket crashes

"Scott E. Preece" <preece%ccvaxa@GSWD-VMS.ARPA>
Mon, 8 Sep 86 09:54:06 cdt

> From: mogul@decwrl.DEC.COM (Jeffrey Mogul)

> I don't often shop at that market, partly because the markets I do use
> have cashiers who know what things are rather than relying on the
> computer. Some day, just for fun, I might mark a pound of pecans with
> the code number for walnuts, and see if I can save some money.

Does the word "fraud" mean anything to you?

Even if your pet cashier can tell at sight a pound of peanuts from a pound of walnuts, I don't see any reason to assume he would know what the correct price of either was or even which was more expensive on a particular day. The cashier is just as dependent on price stickers in a piece marked store as the scanner is on the UPC label in a scanner store.

If I were designing a cash register, I'd make sure it could retain the current session through a power outage (no re-ringing the stuff already in the bags), but I don't think I'd require it to work while the power was off. Personally, if I were in a store when the power went out, I would leave quickly. If power loss is COMMON in the area where the store is built, the designers should work around it (perhaps by providing battery-powered scanners or emergency backup power); in my neighborhood I think it's reasonable to write off as a minor inconvenience -- the speed and efficiency of the scanners when the power is on is a more than reasonable trade for the inconvenience the tiny part of the time it isn't.

--

scott preece
gould/csd - urbana
uucp: ihnp4!uiucdcs!ccvaxa!preece
arpa: preece@gswd-vms

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 54

Monday, 15 September 1986

Contents

- Ada Inherently Secure?
  Mike McLaughlin
- A million lines of code works the first time?
  Ken Calvert
- Computers and Ethics
  Mark S. Day
- New book: HUMAN RELIABILITY: With Human Factors
  Elizabeth ?
- Answers to WWMCCS Intercomputer Network questions
  Harold E. Russell
- Info on RISKS (comp.risks)

Ada Inherently Secure?

Mike McLaughlin <mikemcl@nrl-csr>
Fri, 12 Sep 86 09:14:09 edt

The 8 September 1986 issue of InformationWEEK carried an article "Ada Goes to Work." A box in that article is "Ada is Finding More Job Opportunities With European Telecommunications and Banking Corporations" by Philip Hunter. The following excerpts are from the box. Deletions... (bridges) [comment]:

"... The Finnish bank Kansallis Osake Pankki has standardized on Ada for some... systems, having decided that the language is much better than Cobol for developing secure fail-safe applications, with sound structure and strong management control."

"... Barclays privately admits that Ada could be the logical successor to Cobol for financial systems where security and fail-safe operation are essential, ..."

"... its chief appeal to banks is the rigorous structure... This prevents individual(s)... from making changes that affect other parts of the system. The ... application is then to a large extent shielded both from careless coding... and from deliberate tampering-including the insertion of logic time [sic] bombs... "

"... Ada... helps project managers construct secure reliable systems."

[several paragraphs omitted]

"... British Petroleum and Shell... are evaluating its use for telemetry, ... The... Schlumberger group has... standardize(d) on Ada for oil-field simulation systems, ..."

"Corporations here in the (U.S.) also are taking up Ada for simulation applications, but Europe is way ahead in use (of Ada) for telecommunications, ..."

"(other uses include)... Computer Integrated Manufacturing, where a universal applications-programming environment is needed ... to drive a variety of devices, such as robots, machine tools, and vision systems. "

[I have left out several concluding paragraphs. The thrust of the article (Ada doing fine in Europe) is skewed by my selection of matters relating to safety, security, and reliability.]

- Mike McLaughlin <mikemcl@nrl-csr.arpa>

A million lines of code works the first time?

Ken Calvert <calvert@sally.utexas.edu.UTEXAS.EDU>
Fri, 12 Sep 86 11:52:37 cdt

Heard on NPR's "All Things Considered" yesterday evening:
An Air Force Lt. Col., speaking about a kinetic energy weapons test earlier this week, which apparently went better than expected in several respects. If this isn't an exact quote (I heard it twice, but didn't write it down at the time), it's real close:
"We wrote about a million lines of new computer code, and tested them all for the first time, and they all worked perfectly."

"Interesting if true - and interesting anyway." - Mark Twain.

Ken Calvert
Univ. of Texas Computer Sciences

Computers and Ethics

Mark S. Day <MDAY@XX.LCS.MIT.EDU>
Thu 11 Sep 86 09:48:37-EDT

In a recent issue of Risks, a contributor suggested the possibility of substituting the UPC code for walnuts on a package of pecans, to "save some money". While I am fairly sure that person was joking, it does point out an interesting phenomenon in the area of computer-related risks. That is, as soon as a computer is involved, people seem more willing to commit acts of fraud, theft, and espionage than they would in the absence of a computer. Thus, people will talk about switching UPC price tags who would view switching non-computerized price tags as fraud. Similarly, people will read mail and data files stored on a timesharing system, even though it's unacceptable to rifle through people's desks.

I don't believe that this is due to inadequate security measures on computers. My desk is unlocked, but that hardly constitutes license for people to paw through it, even in my absence. Two possible explanations that occur to me are

1) Novelty -- computers are sufficiently new that they haven't been included in people's "social conditioning". All of the little stories that tell children not to steal, not to lie, etc. don't seem to apply to computers and bits.

2) Distance -- computers serve as intermediaries distancing the perpetrator from the victim. It is easier to consider and carry out unethical actions when they appear to be carried out on a machine rather than a person.

What, if anything, can/should be done about this problem?

--Mark

New book: HUMAN RELIABILITY: With Human Factors

<ELIZABETH%OZ.AI.MIT.EDU@XX.LCS.MIT.EDU>
Wed, 10 Sep 1986 13:47 EDT

A blurb from Pergamon Press came in the mail today; I thought RISKS readers might be interested in this book.

Title: _HUMAN_RELIABILITY:_With_Human_Factors_
Author: Balbir S. Dhillon, Mechanical Engineering/University of Ottawa
Other: 1986; 272 pp.; softcover 24.50, harcover 43.50

Blurb: "This first-of-its-kind text explains the important role people play in the overall reliability of engineering systems, since various systems are interconnected by human links. Detailed coverage of these systems and links are given through data collection and analysis, development of reliability prediction methods and techniques, and numerous ready-to-use formulas and mathematical models for predicting human reliability in a variety of situations. The introductory material eliminates the need for prior knowledge of mathematics and reliability. Exercises and references follow each chapter.

"Designed for upper-level undergraduate and graduate students, this
text will find application across many disciplines since human error
is a common problem..."

AAAI-86 Report, RISKS 3.41

Harold E. Russell <russell@mitre.ARPA>
Thu, 11 Sep 86 14:25:48 -0500

The report from AAAI-86 (RISKS 3.41, Alan Wexelblat, wex@mcc.arpa) had two questions (Q7 & Q8) relating to the WWMCCS Intercomputer Network (WIN).

The WIN, which is the communications component of WWMCCS, has received a great deal of bad press dating from the period 1977-79. Some of it may be pertinent to RISKS FORUM.

RISK: Using obsolete data for system evaluation.

The most vociferous complaints about WIN date from the period 1977-79 which was a transition phase from prototype (PWIN) to operational status. Use of data from that period may be of academic interest but it is not relevant to the present WIN which has current technology hardware and vastly improved software. I visited several WWMCCS sites after the transition and found satisfied users who were doing things that they considered impractical a few years before. In some cases, the WIN was outperforming every other communications medium to the point of operating where the parallel communication channels failed or were hopelessly saturated. WIN is now handling more data and serving more users than was originally anticipated. There are still people whose contempt for WIN is based on data from the transition era.

RISK: Premature transition from prototype to operational status.

Transitioning from a prototype to production or operational status is always a calculated risk. This was no different in the case of the WIN. Go ahead was given based IN PART on the following:

1. There were still minor but correctable technical flaws in the WIN.
2. Even in its imperfect state, the WIN provided capabilities which were not otherwise available.
3. A situation existed where no applications software was being developed for WIN because WIN was not yet available for development of applications software.
4. There would be a learning curve for the applications development people where the remaining WIN technical problems could be resolved before the learning curve started to rise significantly.
5. There was no way of economically or effectively modeling or testing the full-blown military network.
6. Certain categories of highly sensitive military messages would be prohibited in the WIN. No reliance would be placed on an unproven

system.

In the case of the WIN, the gamble paid off handsomely, but there are still numerous criticisms from people who could not or would not understand the situation that existed in the late 1970s.

RISK: Adaptation of technology from a different environment.

The WIN was directly derived from the purportedly highly successful ARPANET which dated back to the late 1960s. The ARPANET of that era was essentially a heterogeneous network linking universities and government research houses. There were however flaws in the network architecture and implementation that were unknown, unrecognized or otherwise not recorded, which came to light in the homogeneous military environment. No one much knew or cared if the University of West Academia unexpectedly dropped out of the network because of failures in home-grown software or hardware. In the WIN, a lot of people will take notice if the Pentagon suddenly drops out of the network. Much of the development effort and many of the problems reported in the 1977-79 period were associated with correcting deficiencies in the ARPANET architecture and implementation. The ARPANET was and still is a very good research network where problems are analyzed and corrected on a time-, money-, and talent-available basis. There may be serious problems in the wholesale transfer of laboratory technology to other environments especially critical large-scale military installations.

RISK: Becoming a victim of one's own success.

At well-managed and well-run sites the WWMCCS/WIN provides good service and reliability to those who understand its capabilities and limitations. This results in a good reputation which causes the demands for service extension to new users beyond those originally intended or causes existing users to increase their utilization of the system. Failure to accommodate these demands yields criticisms of poor response and inadequate support. In order to support more users or increased utilization, the site equipment would probably require additional hardware which is difficult to formally justify and fund. At the present time a typical WWMCCS site has less than half the equipment that the vendor defines as a maximum hardware configuration. If more users are granted access than the equipment can support, then performance can be expected to degrade and complaints to increase. The WIN provides solid, reliable, effective communications among the WWMCCS sites for file transfer, teleconferencing, remote terminal access, and mail, but it has throughput limitations. Performance tests, which I conducted two years ago, showed that minimal WWMCCS ADP computers are capable of driving the communications lines at near theoretical capacity. Some people understand why their M16 rifle can't shoot 10 miles but will not be convinced that it takes a while to transfer a megabyte file over a 56K baud communications link.

WWMCCS ADP and the WIN have a lot of room for technical improvement. However, the biggest problems are not technical, but government regulations, redtape, funding, and retention of trained, capable personnel. The continual references to statistics and data nearly a decade old is misleading and masks current problems and issues.

As always, these opinions may not reflect those of my employers, associates, or customers, past or present.

Search RISKS using swish-e

Report problems with the web pages to the maintainer

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 55

Monday, 15 September 1986

Contents

- Hardware/software interface and risks
  - Kevin Kenny
- F-16
  - Holleran
  - Ken Miya
  - Ihor Kinal
  - Doug Wade
- Info on RISKS (comp.risks)

Hardware/software interface and risks

Kevin Kenny <kenny@b.cs.uiuc.edu>
Thu, 11 Sep 86 10:37:05 CDT

In RISKS 3.53 mlbrown at ncsc-wo writes:

>In RISKS 3.51 Bill Janssen writes of errors made in failing to consider
>the interaction of the hardware and the software under design. This
>failing was all too common in the writing of assembly and machine code
>during the early days of programming. Discrete wired machines often had
>OP codes that were not generally well known (i.e. the computer designers
>kept it secret)...

This posting raises another interesting issue; in any system with a long service life, there is a likelihood that the underlying hardware technology will change. Use of anything undocumented on a particular machine is asking for trouble when that machine is replaced with a ``compatible'' one that lacks the undocumented feature.

In fact, the undocumented op-codes on 4-pi and 360 were not ``kept secret'' by the machine designers; in many cases they simply were not foreseen. It turned out that the combinations of operations that were performed by certain bit patterns did something useful. The modern microprocessors have this tendency also; witness the plethora of undocumented opcodes on the Z80.

The modern mainframe manufacturers have all been burned at one time or another by users who take advantage of undocumented features and then have their programs fail when transported to a ``compatible'' machine using newer technology; the IBM 1401 compatibles brought out by Honeywell after IBM dropped the product line are the classic example. Some of the manufacturers now consider it worth the cost to add logic to verify that a program is using only documented instructions (generate a machine fault rather than an undocumented result); their experience is that documenting something to be forbidden doesn't keep the hackers from using it. There's some justification for the ``everything not permitted is forbidden'' attitude; I've seen mysterious failures years after a machine conversion caused by hardware incompatibilities in little-used areas of the software.

I have also discovered successful penetrations of security on systems in which undocumented opcodes allowed user programs to perform privileged operations. I will deliberately refrain from discussing these further since some of the designs thus penetrated are still in service in the field.

The goal that the hardware designers should aim for is to provide predictable results under all circumstances, even the cases that are documented to be illegal.

Kevin Kenny

F-16 exceeding 55 mph

<Holleran@DOCKMASTER.ARPA>
Thu, 11 Sep 86 00:49 EDT

I would like to provide some diversion on Jon Jacky's comments.

> >Date: Mon, 8 Sep 86 16:55:19 PDT
> >From: jon@uw-june.arpa (Jon Jacky)
> >Subject: Upside-down F-16's and "Human error"
>
> >... should prevent any unreasonable behavior. This way lies
> >madness. Do we have to include code to prevent the speed from
> >exceeding 55 mph while taxiing down an interstate highway?

I agree with this and subsequent statements about the capabilities of the operator (the pilot).

Let's examine a silly analysis of providing that particular code. After you code the routine to prevent the " exceeding the speed", you are going to have to test it. Thus, the F-16 will have to "attempt" to exceed 55 mph on the expressway. Whether the code is there or not, the trooper is still going to give you a ticket. You have already made his day, but no one will believe him without the pilot getting a ticket. Besides he has to make his quota. So you may as well save your money for more important coding. Then the pilot will appear on either 60 minutes or Johnny Carson to explain his side of the problem. The analysis could go further but it belongs in a comedian's dialogue now.

I would say that many "unreasonable behavior" situations being analyzed in a silly mode would show that some coding efforts should not be done. You may find out that certain situations cannot be tested in a justifiable fashion. As Jon Jacky and others have concluded, lets be reasonable in the questions responsible people should be addressing vice situations which have little chance of occuring. Good analysis will be better if common sense helps us to priortize these situations.

Re: F-16 software

Eugene Miya <eugene@AMES-NAS.ARPA>
10 Sep 1986 1317-PDT (Wednesday)

It seems F-16's are a hot topic everywhere. I think it's novelty thing like computers except for aeronautics.

> I am trying to make the point that the gross simplification of
> "preventing bomb release while inverted" doesn't map very well to what I
> assume the actual goal is: "preventing weapons discharge from damaging
> the aircraft". This is yet another instance where the assumptions made
> to simplify a real-world situation to manageable size can easily lead to
> design "errors", and is an architypical "computer risk" in the use of
> relatively simple computer models of reality.
>
> Wayne Throop <the-known-world>!mcnc!rti-sel!dg_rtp!throopw

Excellent point.

Several things strike me about this problem. First, the language used by writers up to this point don't use words like "centrifugal force" and "gravity." This worries me about the training of some computer people for jobs like writing mission critical software [Whorf's "If the word does not exist, the concept does not exist."] I am awaiting a paper by Whitehead whch I am told talks about some of this.

It can certainly be acknowledged that there are uses which are novel (Spencer cites "lob" bombing, and others cite other reasons [all marginal]) equal concern must be given to straight-and-level flight AND those novel cases. In other words, we have to assume some skill on the part of pilots [Is this arrogance on our part?].

Another problem is that planes and Shuttles do not have the types of sensory mechanisms which living organisms have. What is damage if we cannot "sense it?" Sensing equipment costs weight. I could see some interesting dialogues ala "Dark Star."

Another thing is that the people who write simulations seem to have the

great difficulty discriminating between the quality of thier simulations
and "real world" in the presence of incomplete cues (e.g., G-forces,
visual cues, etc.) when solely relying on things like instrument disk
[e.g., pilot: "Er, you notice that we are flying on empty tanks?" disturbed
pilot expression, programmer: "Ah, it's just a simulation."]
Computer people seem to be "ever the optimist." Besides, would you ever
get into a real plane with a pilot who's only been in simulators?

Most recently, another poster brought up the issue of autonmous weapons.
We had a discussion of of this at the last Palo Alto CPSR meeting.
Are autonmous weapons moral? If an enemy has a white flag or hand-ups,
is the weapon "smart enough" to know the Geneva Convention (or is too
moral for programmers of such systems)?

On the subject of flight simulators: I visited Singer Link two years
ago (We have a DIG 1 system which we are replacing). I "crashed" underneath
the earth and the polygon structure became more "visible." It was like
being underneath Disneyland.

--eugene miya sorry for the length, RISKS covered alot.
NASA Ames Research Center
President
Bay Area ACM SIGGRAPH

Re: RISKS-3.53

<cbosgd!mtung!ijk@ucbvax.Berkeley.EDU>
Fri, 12 Sep 86 07:44:24 PDT

Mike Brown wrote:

<> Its fine to tell the pilot "Lower your wheels before you land, not
<> after" but we still have gear up landings. We should not concern ourselves
<> with checking for and preventing any incorrect behavior but we should preclude
<> that behavior which will result in damage to or loss of the aircraft or the
<> pilot. We do not need to anticipate every possible mistake that he can make
<> in this regard either - all we need to do are to identify the hazardous operational modes and prevent
<> their occurrence.

I disagree that software MUST prevent: what about the case when an
aircraft can lower only ONE side of its landing gear???? A belly-up
landing is then the only way to go [assume combat damage, or something,
so that the pilot can't eject, and the computer INSISTS on lowering
the landing gear whenever you attempt to go under 50 feet, or
something stupid like that].

On the other hand, some of the latest experimental planes are
totally UNFLYABLE by normal human control -- for those planes,
the software better be reliable, because there is no backup!!!

Obviously, one can present arguments for each side [human vs computer
having the last say -- at TMI, computers were right, but ...] I

would say that if humans do override CRITICAL computer control [like
TMI], then some means of escalating the attention level must be
invoked [e.g., have the computers automatically notify the NRC].
Again, there's lots of tradeoffs to be made [seriousness of the problem,
timeliness of the response necessary, etc.] which means thats there's
NO PAT answer in most cases, just hope that people involved in these
cases realize the possible consequences of their work. In that
case one could argue for professional certification in these fields
[we're software ENGINEERS, right?!? : you wouldn't to go over a bridge
built by an uncertified mechanical enginerr, would you?? What if the
software he used was written by a flake?]; if not certification,
then perhaps the software should undergo wide scrutiny by independent
evaluators [I'd feel a lot better if I knew that the software controlling
nuclear plants had undergone such scrutiny].

Enough said, I believe.
Ihor Kinal
ihnp4!mtung!ijk.

---

re. F-16 Software.

<Doug_Wade%UBC.MAILNET@MIT-MULTICS.ARPA>
Wed, 10 Sep 86 11:42:14 PDT

Reading comments about putting restraints on jet performance within
the software reminded me of a conversation I had a few years ago
at an air-show.
In talking to a pilot who flew F-4's in Vietnam he mentioned that
the F-4 specs said a turn exerting more than say 8 G's would cause
the wings to "fall off". However in avoiding SAMs or ground-fire
they would pull double? this with no such result.
My comment to this, is what if a 8G limit had been programmed into
the plane (if it had been fly-by-wire). Planes might have been hit and
lost which otherwise were saved by violent maneuvers. With a SAM targeted
on your jet, nothing could be lost by exceeding the structural limitations
of the plane since it was a do-or-die situation.
I'm sure 99.99% of the lifetime of a jet is spent within designed
specifications, but should software limit the plane the one time
a pilot needs to override this constraint?

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 56

Tuesday, 16 September 1986

Contents

- Massive UNIX breakins at Stanford
  Brian Reid
- Info on RISKS (comp.risks)

Massive UNIX breakins at Stanford

Brian Reid <reid@decwrl.DEC.COM>
16 Sep 1986 1519-PDT (Tuesday)

Lessons learned from a recent rash of Unix computer breakins

Introduction

A number of Unix computers in the San Francisco area have recently been plagued with breakins by reasonably talented intruders. An analysis of the breakins (verified by a telephone conversation with the intruders!) show that the networking philosophy offered by Berkeley Unix, combined with the human nature of systems programmers, creates an environment in which breakins are more likely, and in which the consequences of breakins are more dire than they need to be.

People who study the physical security of buildings and military bases believe that human frailty is much more likely than technology to be at fault when physical breakins occur. It is often easier to make friends with the guard, or to notice that he likes to watch the Benny Hill show on TV and then wait for that show to come on, than to try to climb fences or outwit burglar alarms.

Summary of Berkeley Unix networking mechanism:

The user-level networking features are built around the principles of "remote execution" and "trusted host". For example, if you want to copy a file from computer A to computer B, you

type the command
    rcp A:file B:file
If you want to copy the file /tmp/xyz from the computer that you are now using over to computer C where it will be called /usr/spool/breakin, you type the command
    rcp /tmp/xyz C:/usr/spool/breakin
The decision of whether or not to permit these copy commands is based on "permission" files that are stored on computers A, B, and C. The first command to copy from A to B will only work if you have an account on both of those computers, and the permission file stored in your directory on both of those computers authorizes this kind of remote access.

Each "permission file" contains a list of computer names and user login names. If the line "score.stanford.edu reid" is in the permission file on computer "B", it means that user "reid" on computer "score.stanford.edu" is permitted to perform remote operations such as rcp, in or out, with the same access privileges that user "reid" has on computer B.

How the breakins happened.

One of the Stanford campus computers, used primarily as a mail gateway between Unix and IBM computers on campus, had a guest account with user id "guest" and password "guest". The intruder somehow got his hands on this account and guessed the password. There are a number of well-known security holes in early releases of Berkeley Unix, many of which are fixed in later releases. Because this computer is used as a mail gateway, there was no particular incentive to keep it constantly up to date with the latest and greatest system release, so it was running an older version of the system. The intruder instantly cracked "root" on that computer, using the age-old trojan horse trick. (He had noticed that the guest account happened to have write permission into a certain scratch directory, and he had noticed that under certain circumstances, privileged jobs could be tricked into executing versions of programs out of that scratch directory instead of out of the normal system directories).

Once the intruder cracked "root" on this computer, he was able to assume the login identity of everybody who had an account on that computer. In particular, he was able to pretend to be user "x" or user "y", and in that guise ask for a remote login on other computers. Sooner or later he found a [user,remote-computer] pair for which there was a permission file on the other end granting access, and now he was logged on to another computer. Using the same kind of trojan horse tricks, he was able to break into root on the new computer, and repeat the process iteratively.

In most cases the intruder left trojan-horse traps behind on every computer that he broke into, and in most cases he created login accounts for himself on the computers that he broke into. Because no records were kept, it is difficult to tell exactly how many machines were penetrated, but the number could be as high as

30 to 60 on the Stanford campus alone. An intruder using a similar modus operandi has been reported at other installations.

How "human nature" contributed to the problem

The three technological entry points that made this intrusion possible were:

* The large number of permission files, with entirely too many permissions stored in them, found all over the campus computers (and, for that matter, all over the ARPAnet).

* The presence of system directories in which users have write permission.

* Very sloppy and undisciplined use of search paths in privileged programs and superuser shell scripts.

Permissions: Berkeley networking mechanism encourages carelessness.

The Berkeley networking mechanism is very very convenient. I use it all the time. You want to move a file from one place to another? just type "rcp" and it's there. Very fast and very efficient, and quite transparent. But sometimes I need to move a file to a machine that I don't normally use. I'll log on to that machine, quickly create a temporary permission file that lets me copy a file to that machine, then break back to my source machine and type the copy command. However, until I'm quite certain that I am done moving files, I don't want to delete my permission file from the remote end or edit that entry out of it. Most of us use display editors, and oftentimes these file copies are made to remote machines on which the display editors don't always work quite the way we want them to, so there is a large nuisance factor in running the text editor on the remote end. Therefore the effort in removing one entry from a permission file--by running the text editor and editing it out--is high enough that people don't do it as often as they should. And they don't want to *delete* the permission file, because it contains other entries that are still valid. So, more often than not, the permission files on rarely-used remote computers end up with extraneous permissions in them that were installed for a one-time-only operation. Since the Berkeley networking commands have no means of prompting for a password or asking for the name of a temporary permission file, everybody just edits things into the permanent permission file. And then, of course, they forget to take it out when they are done.

Write permission in system directories permits trojan horse attacks.

All software development is always behind schedule, and programmers are forever looking for ways to do things faster. One convenient trick for reducing the pain of releasing new versions of some program is to have a directory such as /usr/local/bin or /usr/stanford/bin or /usr/new in which new or locally-written versions of programs are kept, and asking users to put that directory on their search paths. The systems programmers then give themselves write access to that directory, so that they can intall a new version just by typing "make install" rather than taking some longer path involving root permissions. Furthermore, it somehow seems more secure to be able to install new software without typing the root password. Therefore it is a nearly-universal practice on computers used by programmers to have program directories in which the development programmers have write permission. However, if a user has write permission in a system directory, and if an intruder breaks into that user's account, then the intruder can trivially break into root by using that write permission to install a trojan horse.

Search paths: people usually let convenience dominate caution.

Search paths are almost universally misused. For example, many people write shell scripts that do not specify an explicit search path, which makes them vulnerable to inheriting the wrong path. Many people modify the root search path so that it will be convenient for systems programmers to use interactively as the superuser, forgetting that the same search path will be used by system maintenance scripts run automatically during the night. It is so difficult to debug failures that are caused by incorrect search paths in automatically-run scripts that a common "repair" technique is to put every conceivable directory into the search path of automatically-run scripts. Essentially every Unix computer I have ever explored has grievous security leaks caused by underspecified or overlong search paths for privileged users.

Summary conclusion: Wizards cause leaks

The people who are most likely to be the cause of leaks are the wizards. When something goes wrong on a remote machine, often a call goes in to a wizard for help. The wizard is usually busy or in a hurry, and he often is sloppier than he should be with operations on the remote machine. The people who are most likely to have permission files left behind on stray remote machines are the wizards who once offered help on that machine. But, alas, these same wizards are the people who are most likely to have write access to system directories on their home machines, because it seems to be in the nature of wizards to want to collect as many permissions as possible for their accounts. Maybe that's how they establish what level of wizard that they are. The net result is that there is an abnormally high probability that when an errant permission file is abused by an intruder, that it will lead to the account of somebody who has an unusually large collection of permissions on his own machine, thereby making it easier to break into root on that machine.

Conclusions.

My conclusions from all this are these:

* Nobody, no matter how important, should have write permission into any directory on the system search path. Ever.

* Somebody should carefully re-think the user interface of the Berkeley networking mechanisms, to find ways to permit people to type passwords as they are needed, rather than requiring them to edit new permissions into their permissions files.

* The "permission file" security access mechanism seems fundamentally vulnerable. It would be quite reasonable for a system manager to forbid the use of them, or to drastically limit the use of them. Mechanized checking is easy.

* Programmer convenience is the antithesis of security, because it is going to become intruder convenience if the programmer's account is ever compromised. This is especially true in setting up the search path for the superuser.

Lament

I mentioned in the introduction that we had talked to the intruders on the telephone. To me the most maddening thing about this intrusion was not that it happened, but that we were unable to convince any authorities that it was a serious problem, and could not get the telephone calls traced. At one point an intruder spent 2 hours talking on the telephone with a Stanford system manager, bragging about how he had done it, but there was no way that the call could be traced to locate him. A few days later, I sat there and watched the intruder log on to one Stanford comptuer, and I watched every keystroke that he typed on his keyboard, and I watched him break in to new directories, but there was nothing that I could do to catch him because he was coming in over the telephone. Naturally as soon as he started to do anything untoward I blasted the account that he was using and logged him off, but sooner or later new intruders will come along, knowing that they will not be caught because what they are doing is not considered serious. It isn't necessarily serious, but it could be. I don't want to throw such people in jail, and I don't want to let them get away either. I just want to catch them and shout at them and tell them that they are being antisocial.

Brian Reid
DEC Western Research and Stanford University

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 57

Tuesday, 16 September 1986

Contents

- Computers and the Stock Market (again)
  Robert Stroud
- The Old Saw about Computers and TMI
  Ken Dymond
- Do More Faults Mean (Yet) More Faults?
  Dave Benson
- A critical real-time application worked the first time
  Dave Benson
- Autonomous weapons
  Eugene Miya
- "Unreasonable behavior" and software
  Eugene Miya on Gary Chapman
- Risks of maintaining computer timestamps revisited
  John Coughlin
- Info on RISKS (comp.risks)

Computers and the Stock Market (again)

Robert Stroud <robert%cheviot.newcastle.ac.uk@Cs.Ucl.AC.UK>
Mon, 15 Sep 86 16:53:37 gmt

The computers had a hand in the dramatic fall on Wall Street last week according to an item on the BBC TV news. Apparently, the systems were not designed to cope with the sheer volume of sales, (anybody know more about this?). The report continued

> "In London they still do it the old fashioned way with bits of paper, which makes people think twice before joining in a mindless selling spree. However, all this could change in October with the Big Bang..."

What price progress?

Robert Stroud,

Computing Laboratory,
University of Newcastle upon Tyne.

ARPA robert%cheviot.newcastle@ucl-cs.ARPA
UUCP ...!ukc!cheviot!robert

The Old Saw about Computers and TMI

"DYMOND, KEN" <dymond@nbs-vms.ARPA>
16 Sep 86 09:25:00 EDT

Ihor Kinal says in RISKS-3.55

>Obviously, one can present arguments for each side [human
> vs computer having the last say -- at TMI, computers
>were right, but ...]   I would say that if humans do
>override CRITICAL computer control [like TMI], then
>some means of escalating the attention level must be
>invoked [e.g., have the computers automatically notify
>the NRC].

This belief keeps surfacing but is false. There was no computer control in safety grade systems at TMI -- see the documentation in the Kemeny report and probably elsewhere. There was a computer in the control room but it only drove a printer to provide a hardcopy log of alarms in the sequence in which they occurred. The log is an aid in diagnosing events. The computer (a Bendix G-15 ??) did play a role in the emergency since at one point its buffer became full and something like 90 minutes of alarms were not recorded, thus hampering diagnosis.

On a couple of occasions I have asked NRC people why computers aren't used to control critical plant systems and have been told that "they aren't safety grade." I'm not quite sure what this means, but I take it to mean that computers (and software) aren't trustworthy enough for such safety areas as the reactor protection system. This is not to say that computers aren't used in monitoring plant status, quite different from control.

Ken Dymond
(the opinions above don't necessarily reflect those of my employer
or anybody else, for that matter.)

Do More Faults Mean (Yet) More Faults?

Dave Benson <benson%wsu.csnet@CSNET-RELAY.ARPA>
Sun, 14 Sep 86 19:00:30 pdt

|In RISKS 3.50 Dave Benson comments in "Flight Simulator
|Simulators Have Faults" that
|

| >We need to understand that the more faults found at
| >any stage to engineering software the less confidence one has in the
| >final product. The more faults found, the higher the likelyhood that
| >faults remain.
|
|This statement makes intuitive sense, but does anyone know of any data
|to support this ? Is this true of any models of software failures ?
|Is this true of the products in any of the hard engineering fields -- civil,
|mechanical, naval, etc. -- and do those fields have the confirming data ?
|
|Ken Dymond, NBS

Please read the compendium of (highly readable) papers by M.M.Lehman and L.A.Belady, Program Evolution: Processes of Software Change, APIC Studies in Data Processing No. 27, Academic Press, Orlando, 1985. This provides data. It is (sorry-- should be, but probably isn't) standard in software quality assurance efforts to throw away modules which show a high proportion of the total evidenced failures. The (valid, in my opinion) assumption is that the engineering on these is so poor that it is hopeless to continue to try to patch it up.

Certain models of software failure place increased "reliablity" on software which has been exercised for long periods without fault. One must understand that this is simply formal modelling of the intuition that some faults means (yet) more faults. This is certainly true of all engineering fields. While I don't have the "confirming data" I suggest you consider your car, your friends car, etc. Any good history of engineering will suggest that many designs never are marketed because of an unending sequence of irremediable faults.

The intuitive explaination is: Good design and careful implementation works. This is teleological. We define good design and careful implementation by "that which works".

However, I carefully said "confidence". Confidence is an intuitive assessment of reliability. I was not considering the formalized notion of "confidence interval" used in statistical studies. To obtain high confidence in the number of faults requires observing very many errors, thus lowering one's confidence in the product. To obtain high confidence in a product requires observing very few errors while using it.

I found one! (A critical real-time application worked the first time)

Dave Benson <benson%wsu.csnet@CSNET-RELAY.ARPA>
Sun, 14 Sep 86 22:40:21 pdt

Last spring I issued a call for hard data to refute a hypothesis which I, perhaps mistakenly, called the Parnas Hypothesis:
   No large computer software has ever worked the first time.
Actually, I was only interested in military software, so let me repost the challenge in the form I am most interested in:
   NO MILITARY SOFTWARE (large or small) HAS EVER WORKED IN ITS FIRST

OPERATIONAL TEST OR ITS FIRST ACTUAL BATTLE.
Contradict me if you can. (Send citations to the open literature
to benson@wsu via csnet)

Last spring's request for data has finally led to the following paper:
Bonnie A. Claussen, II
VIKING '75 -- THE DEVELOPMENT OF A RELIABLE FLIGHT PROGRAM
Proc. IEEE COMPSAC 77 (Computer Software & Applications Conference)
IEEE Computer Society, 1977
pp. 33-37

I offer some quotations for your delictation:

The 1976 landings of Viking 1 and Viking 2 upon the surface of
Mars represented a significant achievement in the United States
space exploration program. ... The unprecented success of the Viking
mission was due in part to the ability of the flight software
to operate in an autonomous and error free manner. ...
Upon separation from the Oribiter the Viking Lander, under autonomous
software control, deorbits, enters the Martian atmosphere,
and performs a soft landing on the surface. ... Once upon the surface,
... the computer and its flight software provide the means by
which the Lander is controlled. This control is semi-autonomous
in the sense that Flight Operations can only command the Lander
once a day at 4 bit/sec rate.

(Progress occured in a NASA contract over a decade ago, in that)

In the initial stages of the Viking flight program development,
the decision was made to test the flight algorithms and determine
the timing, sizing and accuracy requirements that should be
levied upon the flight computer prior to computer procurement.
... The entire philosophy of the computer hardware and
software reliability was to "keep it simple." Using the
philosophy of simplification, modules and tasks tend toward
straight line code with minium decisions and minimum
interactions with other modules.

(It was lots of work, as)

When questioning the magnitude of the qulity assurance task,
it should be noted that the Viking Lander flight program development
required approximately 135 man-years to complete.

(But the paper gives no quantitative data about program size or complexity.)

Nevertheless, we may judge this as one of the finest software engineering
acomplishments to date. The engineers on this project deserve far more
plaudits than they've received. I know of no similar piece of software
with so much riding upon its reliable behavior which has done so well.
(If you do, please do tell me about it.)

However, one estimates that this program is on the order of kilolines of FORTRAN
and assembly code, probably less than one hundred kilolines. Thus

Parnas will need to judge for himself whether or not the Viking Lander flight software causes him to abandon (what I take to be) his hypothesis about programs not working the first time.

It doesn't cause me to abandon mine because there were no Martians shooting back, as far as we know...

David B. Benson, Computer Science Department, Washington State University, Pullman, WA 99164-1210 csnet: benson@wsu

---

Autonomous weapons

<LIN@XX.LCS.MIT.EDU>
Tue, 16 Sep 1986 08:31 EDT

From: eugene at AMES-NAS.ARPA (Eugene Miya)

... another poster brought up the issue of autonmous weapons.
We had a discussion of of this at the last Palo Alto CPSR meeting.
Are autonmous weapons moral? If an enemy has a white flag or hand-ups,
is the weapon "smart enough" to know the Geneva Convention (or is too
moral for programmers of such systems)?

What do you consider an autonomous weapon? Some anti-tank devices are intended to recognize tanks and then attack them without human intervention after they have been launched (so-called fire-and-forget weapons). But they still must be fired under human control. *People* are supposed to recognize white flags and surrendering soldiers.

"Unreasonable behavior" and software

<LIN@XX.LCS.MIT.EDU>
Tue, 16 Sep 1986 09:01 EDT

From: Gary Chapman

Risks of maintaining computer timestamps revisited

John Coughlin <JC%CARLETON.BITNET@WISCVM.WISC.EDU>
15 Sep 86 12:14:00 EDT

Some time ago I submitted an item to RISKS describing the way in which the CP-6 operating system requires the time to be set manually during every warm or cold boot. The latest release of this OS contains an improvement: in most cases the time need only be manually set on a cold boot. Unfortunately, with this enhancement came an unusual bug.

The timestamp is stored in a special hardware register, which is modified by certain diagnostic procedures run during preventive maintenance. It seems these diagnostic procedures were not modified to reflect the new use put to the timestamp register. As a result, any time a warm boot was performed after PM, the monitor would freak out at the illegal timestamp and mysteriously abort the boot with a memory fault. Until this bug was patched the only fix was to power the computer down, thus clearing the offending value.

Luckily, the PM procedure set the timestamp register to an impossible value, rather than a realistic but incorrect value. Therefore the problem manifested itself in an obvious way, instead of subtly changing the date and time. Of course this was at the cost of having to fix a hung system. This is yet another illustration of the risk of breaking one thing while fixing another.

/jc

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 58

Wednesday, 17 Sept 1986

Contents

- Massive UNIX breakins
  - Dave Curry
  - Brian Reid
- "Atlanta's been down all afternoon"
  - Alan Wexelblat
- F-16 software
  - Herb Lin
- Viking Project
  - Eugene Miya
- Protection of personal information
  - David Chase
- Autonomous Weapons
  - Ken Laws
- Re: computers and petty fraud
  - Col. G. L. Sicherman
- Info on RISKS (comp.risks)

Massive UNIX breakins

Dave Curry <davy@ee.ecn.purdue.edu>
Wed, 17 Sep 86 08:01:03 EST

Brian -

I feel for you, I really do. Breakins can be a real pain in the neck, aside from being potentially hazardous to your systems. And, we too have had trouble convincing the authorities that anything serious is going on. (To their credit, they have learned a lot and are much more responsive now than they were a few years ago.)

I do have a couple of comments though. Griping about the Berkeley networking utilities is well and good, and yes, they do have their problems. However, I think it really had little to do with the initial breakins on your system. It merely compounded an already exisiting breakin several fold.

Two specific parts of your letter I take exception to:

> One of the Stanford campus computers, used primarily as a mail gateway between Unix and IBM computers on campus, had a guest account with user id "guest" and password "guest". The intruder somehow got his hands on this account and guessed the password.

Um, to put it mildly, you were asking for it. "guest" is probably the second or third login name I'd guess if I were trying to break in. It ranks right up there with "user", "sys", "admin", and so on. And making the password to "guest" be "guest" is like leaving the front door wide open. Berkeley networking had nothing to do with your initial breakin, leaving an obvious account with an even more obvious password on your system was the cause of that.

> There are a number of well-known security holes in early releases of Berkeley Unix, many of which are fixed in later releases. Because this computer is used as a mail gateway, there was no particular incentive to keep it constantly up to date with the latest and greatest system release, so it was running an older version of the system.

Once again, you asked for it. If you don't plug the holes, someone will come along and use them. Again Berkeley networking had nothing to do with your intruder getting root on your system, that was due purely to neglect. Granted, once you're a super-user, the Berkeley networking scheme enables you to invade many, many accounts on many, many machines.

Don't get me wrong. I'm not trying to criticize for the sake of being nasty here, but rather I'm emphasizing the need for enforcing other good security measures:

1. Unless there's a particularly good reason to have one, take all "generic" guest accounts off your system. Why let someone log in without identifying himself?

2. NEVER put an obvious password on a "standard" account. This includes "guest" on the guest account, "system" on the root account, and so on.

   Enforcing this among the users is harder, but not impossible. We have in the past checked all the accounts on our machines for stupid passwords, and informed everyone whose password we found that they should change it. As a measure of how simple easy passwords make things, we "cracked" about 400 accounts out of 10,000 in one overnight run of the program, trying about 12 passwords per account. Think what we could have done with a sophisticated attack.

3. FIX SECURITY HOLES. Even on "unused" machines. It's amazing how many UNIX sites have holes wide open that were plugged

years ago. I even found a site still running with the 4.2
distributed sendmail a few months ago...

4. Educate your police and other authorities about what's going on. Invite them to come learn about the computer. Give them an account and some documentation. The first time we had a breakin over dialup (1982 or so), it took us three days to convince the police department that we needed the calls traced. Now, they understand what's going on, and are much quicker to respond to any security violations we deem important enough to bring to their attention. The Dean of Students office is now much more interested in handling cases of students breaking in to other students' accounts; several years ago their reaction was "so what?". This is due primarily to our people making an effort to educate them, although I'm sure the increased attention computer security has received in the media (the 414's, and so on) has had an effect too.

--Dave Curry
Purdue University
Engineering Computer Network

Massive UNIX breakins

Brian Reid <reid@decwrl.DEC.COM>
17 Sep 1986 0729-PDT (Wednesday)

The machine on which the initial breakin occurred was one that I didn't even know existed, and over which no CS department person had any control at all. The issue here is that a small leak on some inconsequential machine in the dark corners of campus was allowed to spread to other machines because of the networking code. Security is quite good on CSD and EE machines, because they are run by folks who understand security. But, as this episode showed, that wasn't quite good enough.

"Atlanta's been down all afternoon" (!?)

Alan Wexelblat <wex@mcc.com>
Wed, 17 Sep 86 14:38:59 CDT

Last Friday, we attempted to phone (ATT) long distance to Atlanta. After two hours of busy signals we finally decided to try and reach the Atlanta operator. She said that Atlanta had been "down all afternoon."

Does anyone have any info about this?

Alan Wexelblat
ARPA: WEX@MCC.ARPA or WEX@MCC.COM
UUCP: {seismo, harvard, gatech, pyramid, &c.}!ut-sally!im4u!milano!wex

F-16 software

<LIN@XX.LCS.MIT.EDU>
Tue, 16 Sep 1986 17:43 EDT

I spoke to an F-16 flight instructor about this business concerning bomb release when the plane is upside down. He said the software OUGHT to prevent such an occurrence. When the plane is not at the right angle of attack into the air stream, toss-bombing can result in the bomb being thrown back into the airplane.

Re: RISKS-3.57 Viking Project

Eugene Miya <eugene@AMES-NAS.ARPA>
16 Sep 1986 2213-PDT (Tuesday)

Sorry Dr. Benson, I wish to correct you on several points. First off, NASA is the CIVILIAN space agency. NASA takes great pains to emphasize this. We are frequently accused of being puppets of the military and we cannot deny that the DOD are customers and joint researchers, but the DOD also causes us problems. Many scientists in NASA (myself included) work here to try an benefit ALL mankind.

The Viking Project, in particular, is not a military project and the scientists that I know such as Conway Snyder and others would take great offense to your implication. (I think Sagan would be amused and offended, too.)
I can tell you there were bugs in the program. Not
all was perfect. Note the mission had redundency built into it.

What I can tell you about the physical systems is that spacecraft memories at that period of time were very small and quite crude. We are talking hundreds of words of storage not K. We are not talking sophisticated programming either (more like hard coded routines). We are not talking FORTRAN except for the trajectory and orbit determination programs (still in use with 400K to 1M lines of code: Univac FORTRAN V and now VAX VMS FORTRAN). This code may be purchased from COSMIC (I think something like $2K I can look this up). Regarding other project documentation about the nature of the Viking computers and their software, this is all in the public domain in the form of NASA TRs. (Don't ask for all, we are talking TONS of documentation, you want to ask for specifics. and I might be able to help a little [emphasis] by giving you contacts to phone at JPL).

(Un)happily? no Martians shot at the Landers. I don't know how we would have faired. The system had no AI, it's really was not a concurrent system, it had strictly local real-time processing, but not by choice (one-way signal time to Mars is 7 minutes).

Valhalla: that place where Viking Project members go to retire.

--eugene miya

ex-Voyager Program member
ex-JPL/CIT
NASA Ames Research Center

Protection of personal information

David Chase <rbbb@rice.edu>
Tue, 16 Sep 86 23:37:47 CDT

A friend of mine attending a large state university is preparing to interview for jobs. At this university the powers that bureaucratically be "require" that you fill out a form that among other things has your social security number and a statement that (if signed) authorizes release of transcript to people who might wish to employ you. Other things on this form include percentage of college expenses earned, and similar rot that one might wish to keep private. No form, then no on campus interviews.

Just to make things interesting, they wish to place this info in an "experimental" database.

When faced with something like this, what does one person (out of 48000 students, most of them cooperating like sheep) do to get any assurance that private information is not released to undesirables (where the set of "undesirables" is defined by the one person, NOT the university)? This same problem pops up with utilities in this state also, and the bargaining position is even worse than the student's ("I'm sorry sir, but we can't turn on your power until I complete this form, and I can't complete it without your social security number").

Does anyone have any experience with this sort of thing? I read a little blurb while waiting to get my drivers license that told all about how one should most definitely keep one's social security number in confidence, so handing out (without permission) even those 9 digits to an alleged prospective employer is out of line. Never mind that those same 9 digits are your "student number" at this school.

(Perhaps this belongs on Human Nets, but I feel this is a risk--if nothing else, it raises my blood pressure to dangerous levels to hand out private information to pig-headed idiots. I'd also rather prevent some of this now than be the subject of an amusing/shocking anecdote later)

David

Autonomous Weapons

Ken Laws <Laws@SRI-STRIPE.ARPA>
Wed 17 Sep 86 07:10:43-PDT

Eugene Miya asks whether autonomous weapons can be considered moral. Brief thoughts (since Risks may not be the right forum):

Dumb weapons or those guided incompetently are no better -- was the accidentaly bombing of the French Embassy in Libya moral?

Autonomous vehicles (or, for that matter, bombs) are not smart enough to perform trivial civilian duties in cooperative environments (e.g., driving to the grocery store or picking weeds in a corn field). Someday they may be, in which case questions about their intelligence and morality may be worth debating. For now, the assumption is that they are only to be used in situations where anything that moves is a legitimate target and where taking out the wrong "target" is better than taking out no target. This is rather similar to the situation facing nukes, and the moral choices in initiating use are the same. The advantages of autonomous weapons over nukes should be obvious, although there will always be philosophers and humanists who mourn an isolated wrongful death as much as the destruction of a city.

-- Ken Laws

Re: computers and petty fraud

"Col. G. L. Sicherman" <colonel%buffalo.csnet@CSNET-RELAY.ARPA>
Wed, 17 Sep 86 15:33:21 EDT

In RISKS-3.54 Mark S. Day inquires why computerization encourages people to defraud shop clerks.

> ... Thus, people will talk about switching
> UPC price tags who would view switching non-computerized price tags as
> fraud.

This is partly because it's less easily detected. Replacing price tags with bar codes means that the clerk has little or no opportunity to consider whether the price is reasonable. The effect resembles what happened when hand calculators replaced slide rules. By eliminating the element of clerical surveillance, the manager increases efficiency at the cost of security. It's a typical trade-off.

As for the customers ... perhaps the general run of people were never very ethical to begin with?

> Similarly, people will read mail and data files stored on a
> timesharing system, even though it's unacceptable to rifle through
> people's desks.

There are two active changes here. First, a time-sharing system is perceived as a shared facility (even if it runs VM! :-), a commune rather than an apartment house, so to speak. This has been reinforced by the development of message systems. Second, the phenomenal progress in communication in recent years has undermined public support for privacy. The subject of privacy has been vexing and misleading pundit lately; the best treatment of it is to be found in _The Gutenberg

Galaxy_ by H. M. McLuhan. (It's nothing like the typical liberal or illiberal arguments one normally reads.)

A third factor, and I think a significant one, is the re-alignment of popular loyalty. Large societies are products of the age of print. In particular, print provides the inspiration for uniform, stable laws, language, and conventions; it also creates the necessary illusion of commonality by virtue of the physical uniformity of print and the impersonality of publishing. (One could add that large states and countries are perceptible chiefly by virtue of printed maps.) In an age of fast, easy communication, artifacts like countries grow to appear unreal and arbitrary. People are coming to prefer to deal directly with one another, and personal loyalties are out-weighing loyalties to abstractions like country and society. I do not believe that this is a bad thing; it increases strife, but reduces international war.

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 59

Saturday, 20 September 1986

Contents

- Computers and Wall Street
  Robert Stroud
- Report from the Computerized Voting Symposium
  Kurt Hyde
- Computers, TMI, Chernobyl, and professional licensing
  Martin Harriman
- Failsafe software
  Martin Ewing
- Software vs. Mechanical Interlocks
  Andy Freeman
- How Not to Protect Communications
  Geoff Goodfellow
- Info on RISKS (comp.risks)

Computers and Wall Street

Robert Stroud <robert%cheviot.newcastle.ac.uk@Cs.Ucl.AC.UK>
Thu, 18 Sep 86 14:07:59 gmt

I came across an article in Computing which gives more details about the way in which computer systems are influencing the stock market. It suggests that dealers are forced to rely on the "intuition" of their system, even against their better judgement, for fear of being caught out. Personally I find this trend very alarming, but perhaps the fluctuations on the stock market are just "noise" with no lasting influence on the real economy. Unfortunately, the "noise" can be heard around the world.

Robert Stroud,
Computing Laboratory,
University of Newcastle upon Tyne.

ARPA robert%cheviot.newcastle@cs.ucl.ac.uk (or ucl-cs.ARPA)
UUCP ...!ukc!cheviot!robert

"Technology led Wall Street to drop prices" by Alex Garrett

The crash in prices which wiped a record amount off the value of shares on Wall Street last week was largely the result of computerised dealing systems failing to read the market.

Computer generated selling of shares was estimated to account for almost 50% of the transactions that caused a record volume of 240 million shares to change hands last Friday. But it is believed that the effect of the computers was to exaggerate the underlying movement in the market, so that many shares were sold unnecessarily.

The problem has arisen as a number of factors conspired to make the US stock markets subject to increasing fluctuations, which in turn has caused stockbrokers to rely far more heavily upon the split-second advice of their computer systems. In particular, many systems are triggered by a drop in share price to instruct a dealer to sell, and he will often do so, even against his better nature, for fear of being caught out.

.... this kind of feature has yet to be adopted in the UK.

Ian Reid ... said that although shares will often recover their price within a short time, some of the computer systems in the US do not have the intuition to see this.

Report from the Computerized Voting Symposium

Jekyll's Revenge 264-7759 MKO1-2/E02 <hyde%abacus.DEC@decwrl.DEC.COM>
Friday, 19 Sep 1986 11:37:13-PDT

Belated Report from the Symposium on Security and Reliability of Computers in the Electoral Process -- August 14th & 15th, 1986

The participants came from many backgrounds, computer people, writers, attorneys, and even one Secretary of State. Some of the highlights emphasized by one or more speakers were:

- o Lever voting machines are still the fastest way to count votes. The computerized vote counting machines are slower than lever machines, but faster than paper ballots.

- o Lever voting machines still appear to be the safest way to count votes.

- o The State of Illinois tested its computerized voting equipment and found numerous instances of errors in vote counting, primarily in undervotes, overvotes, and straight party crossovers.

NOTE: An undervote is voting for fewer candidates than the maximum allowed for an office. An overvote is voting for more candidates than allowed for an office. A straight party crossover is casting a vote to be applied to all members of a party and then switching one or more votes to candidates from another party.

- o A group of Computer Science students at Notre Dame (South Bend, IN) tested a punch card voting system with a group of test ballots. By altering only the control cards they were able to manage the vote totals to predictable incorrect totals.

Some of the recommendations made by one or more speakers were:

- o Five percent of all votes cast should be recounted by different method than the original count.

- o Security standards for computerized voting are needed immediately. The expanding use of computerized vote counting equipment may preclude an effective implementation of such a standard.

- o Punch card ballots should be redesigned to make the punch card into a ballot that is readable by the voter as well as by the computer.

- o Internal procedures of computerized voting equipment must be open to the public in order to let the public be in control and to assure public confidence in the electoral process.

- o Computerized voting equipment must have the capability of allowing the voter to monitor the ballots cast by the computer to be sure it has voted as instructed.

- o There should be public domain vote counting software in order that companies not have to keep their programs for proprietary ownership reasons.

  NOTE: Does anyone know of a Computer Science student looking for a project? I'm willing to share my notes.

  Is there anyone with the resources to build prototypes that have security features, such as voter-readable punch cards or a computer-generated, recountable ballot?

Bill Gardner, New Hampshire's Secretary of State, informed us that New York City is planning to purchase new voting equipment. This is likely to become a de facto standard for New York State and, possibly, for whole the nation. Risks Forum people who'd like to contact the New York City Task Force should contact:

David Moscovitz

New York City Elections Project
2 Lafayette Street, 6th Floor
New York, NY 10007
(212) 566-2952

The results of my informal poll on trusting a computerized voting system:

	Trust	Not Trust	Undecided
(1) Internal Procedures secret Results not monitored by voter	2/40	38/40	0
(2) Internal Procedures Revealed Results not monitored by voter	6/40	34/40	0
(3) Internal Procedures secret Results can be monitored by voter	10/40	28/40	2/40
(4) Internal Procedures Revealed Results can be monitored by voter	24/40	11/40	5/40

Computers, TMI, Chernobyl, and professional licensing

Martin Harriman <MARTIN%SRUCAD%sc.intel.com@CSNET-RELAY.ARPA>
Wed, 17 Sep 86 09:42 PDT

The NRC does require testing and certification of the software used in the design of nuclear power plants: this includes the software used for seismic simulations, fueling studies, and simulations of coolant behavior (which can get quite complex in BWR designs).

The reactors themselves are designed to be stable, so they do not require a complex control system for safe operation (unlike military aircraft with negative aerodynamic stability). Incidentally, the feedback mechanisms used to produce stability in US reactor designs are missing from graphite moderated, water damped designs like Chernobyl; this lack of stability contributed to the initial explosion at Chernobyl.

Professional licensing is state-regulated; I'm not aware of any states with a professional engineer exam for software engineers. I don't believe that professional licensing is all that useful; I'm more interested in quality assurance for safety-related software (and hardware) than in ensuring that some fraction of the people developing the software passed an examination. It would be fairly amusing if PE registration became popular with software engineers, since it would mean they would all need to learn a fair chunk of civil engineering (the Engineer In Training exam requires it).

--Martin Harriman <martin%srucad@sc.intel.com>

Failsafe software

Martin Ewing <mse%Phobos.Caltech.Edu@DEImos.Caltech.Edu>
Thu, 18 Sep 86 09:57:27 PDT

risks%Phobos.Caltech.Edu@DEImos.Caltech.Edu

How can we even dream of SDI or fly-by-wire aircraft when I just received 12 nearly identical copies of the last ARMS-D mailing, at 33 KB a crack?

Seriously, this is an example of failsafe: if some transmission error occurs before a message transmission is complete, send it again, and again, and again... And no one is even shooting at the net, as far as I know.

Martin Ewing

Software vs. Mechanical Interlocks

Andy Freeman <FREEMAN@SUMEX-AIM.ARPA>
Thu 18 Sep 86 10:16:01-PDT

One current advantage of mechanical interlocks is that they can (usually) be bypassed or modified in the field. If I went on a special toss-bombing mission, I'd be much happier hearing "the mechanical upside-down bomb-release interlock has been removed" than "we just patched out that section of the code and burned a new prom".

-andy

How Not to Protect Communications

the tty of Geoffrey S. Goodfellow <Geoff @ csl.sri.com>
20 Sep 1986 06:52-PDT

[The New York Times, September 13, 1986]

BALTIMORE - The Senate should avoid repeating the mistake made by the House when it unanimously passed the Electronic Communications Privacy Act. Purportedly a benign updating of the 1968 Federal wiretap law designed to guarantee privacy in the electronic age, the bill actually promotes the cellular telephone industry at the expense of the public good.

True enough, obsolete language in the existing wiretap law fails to address digital, video, and other new forms of communications. The proposed law would fix that. But it would also declare certain communications legally private regardless of the electronic medium employed to transport them. The mere act of receiving radio signals, except for certain enumerated services like commercial broadcasts, would become a federal crime.

To disregard the medium is to ignore the essence of the privacy issue. Some media, such as wire, are inherently private. That is, they are hard to get at except by physical intrusion into a residence or up a telephone pole. Others media, notably radio signals, are inherently accessible to the public. Commercial radio and television broadcasts, cellular car telephone transmissions and other "two-way" radio communications enter our homes and pass through our bodies. Cellular phone calls, in fact, can be received by most TV sets in America on UHF channels 80 through 83.

If radio is public by the laws of physics, how can a law of Congress say that cellular communications and other forms of radio are private? The unhappy answer is that the proposed law appears to be a product of technological ignorance or wishful thinking. A similar edict applied to print media would declare newspapers, or portions of them, to be as private as first class mail. The result is plainly absurd and contrary to decades of reasonable legislative and judicial precedent.

In contrast, present Federal statute prescribes a sensible policy for oral communications, protecting only those "uttered by a person exhibiting an expectation that such communication is not subject to interception under circumstances justifying such expectation." To illustrate, a quiet chat in one's parlor would likely be protected. Substitute for the parlor a crowded restaurant or the stage of a packed auditorium, the expectation of privacy is no longer justified. The law would not grant it.

Congress should apply this same logic to electronic communications. The broadcasting of an unencrypted radio telephone call, or anything else, is an inherently public act, whether so intended or not. Thus it violates the "justifiable expectation" doctrine, and warrants no Federal privacy protection.

Protection or no, people will not be stopped from receiving radio signals. Even Representative Robert W. Kastenmeier, Democrat of Wisconsin, who championed the bill in the House, confesses that its radio provisions are essentially unenforceable. They will have no deterrent effect, and they will not increase the privacy of cellular phone calls or other broadcasts. Worse, the act would lull the public into a false presumption of privacy.

On further examination, it appears that the legislation is really more a sham than an honest, if puerile, attempt by Congress to deal with new technology. Its sponsors say they aim to protect all electronic communications equally. Yet the bill sets out at least four categories of phone calls, with varying penalties for interception. Cellular radio calls are guarded by threat of prison, but there is no interdiction whatsoever against eavesdropping on "cordless" telephones of the sort carried around the apartment backyard.

So Congress is about to give the cellular telephone industry ammunition for advertising and bamboozling, promising privacy that does not actually exist. Cellular service companies thereby hope to avoid losing revenue from customers who might use the service less if they understood its vulnerability.

If Congress were serious about privacy in the communications age, it would scrap the Electronic Communications Privacy Act and begin anew. Legislators and the public must first grasp the true properties of new technologies. Are those properties inadequate or unsavory? If so, relief will only come from research and more technology not wishful legislation.

Robert Jesse is a technology consultant. [known to us all as rnj@brl]

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 60

Saturday, 20 September 1986

Contents

- Sanity checks
  Roy Smith
- Viking Flight Software working the `first' time?
  Greg Earle
- A million lines of code works the first time?
  Anonymous
  Dave Benson
  Herb Lin
- Re: Massive UNIX breakins at Stanford
  Scott E. Preece
- Re: Protection of personal information
  Andy Mondore
  Herb Lin
- Announcement of Berkeley Conference on the SDI
  Eric Roberts
- Info on RISKS (comp.risks)

Sanity checks

Roy Smith <allegra!phri!roy@seismo.CSS.GOV>
Fri, 19 Sep 86 16:43:39 edt

I'd like to relate 3 incidents along the lines of people willing to believe anything the computer tells them, what I call the "if it's on green-bar, it it must be true" syndrome.

Incident #1 was two weeks ago. I got 2 items for $5.95 and $8.95 at our local Radio Shack. There was no tax on this sale and I quickly came up with $14.90 in my head (if that's not right, I'm going to be *really* embarrassed). The sales clerk grabbed a calculator and came up with $14.93. I'm not so upset at the fact that he came up with the wrong sum, but that he didn't apply the trivial check that if you have a bunch of numbers, all ending in 0 or 5, the sum must also end in 0 or 5. Moral: Always check your results for sanity and never trust the clerks in Radio Shack.

Incident #2 was a few days later. In a discussion of very large memories I mentioned that 200 bits is the biggest address you would ever need and that 2^200 was about 10^40 (see usenet's net.arch for the past few weeks). How did I come up with that? Easy, I just fired up a desk calculator program, typed "2^200" at it and it typed back "1.70141e+38".

Now, I *knew* this was too small (at 3 or 4 bits per decimal digit I expected about 10^65) so I tried it again. Since it gave the same answer again, I figured it must be right. Of course the problem was overflow (you would think that by now any time I see a Vax print out 1.7e38 a bell would go off in my head). This is even worse than the clerk in Radio Shack; here I had 2 reasons to suspect the answer was wrong and I still blindly believed what the computer told me! Moral: Always check your results for sanity and don't get a big head thinking you're smarter than the clerks at Radio Shack.

Incident #3 was a few years ago. We got a FORTRAN program to predict protein secondary structure (feed it a sequence and it says where it's alpha-form and where it's beta). We fired it up and it ran so we put it into production use. It showed a lot more beta then we expected, but it never occurred to us to suspect the program -- the algorithm was known to slightly over-predict beta and we were perfectly willing to believe that the outrageous amount of beta we were getting was due to that.

To get to the point, the program was from a Vax and we were running it on a pdp-11. The input (3-letter codes) was stored in INTEGER*4's, quitely truncated to INTEGER*2's by the compiler. Most of the codes are distinct in the first 2 letters so this was usually ok. It was, however, turning aspartic acid into asparagine (asp->asn) and glutamic acid into glutamine (glu->gln); both those substitutions tend to result in more beta form! It was weeks before somebody spotted that the annotated sequence the program printed out didn't match the input. Moral #1: Always use sanity checks, but don't blindly rely on them; if your check is "x > y", think before you accept "x <> y" . Moral #2: If the program provides aids like echo printing of input, use them. Moral #3: If you're modifying a program or porting it to a new machine, do regression testing.

Viking Flight Software working the `first' time?

Greg Earle <elroy!smeagol!gorbag!earle@csvax.caltech.edu>
Wed, 17 Sep 86 21:35:44 pdt

Correct me if I am wrong, but for any spacecraft that I know of, virtually every major spacecraft function can be exaustively tested on the ground before the thing ever leaves the pad. About the only thing you can't test (obviously) is the software to actually physically separate the lander from the command module on descent into the atmosphere. Everything else, to my knowledge, can be covered pretty thoroughly. The projects that I am associated with, here at JPL, are involved with test sets that test all the functions of the spacecraft Command Data Subsystem (CDS) which is also called the Payload Data System (PDS) on Mars Observer. In other words, this exercises the flight software that resides in the command data subsystem, and telemetry streams are initiated, commands are uplinked, etc. etc.

Now maybe we want to pick nits and say "Well it worked the first time in Actual Outer Space Usage", which is true, but considering the amount of testing done beforehand (we are now testing breadboard CDS's for missions that won't launch until at least 1991), 'tis not all that surprising when it works ...

Greg Earle UUCP: sdcrdcf!smeagol!earle; attmail!earle
JPL ARPA: elroy!smeagol!earle@csvax.caltech.edu
AT&T: +1 818 354 0876 earle@JPL-MILVAX.ARPA

Anonymous contribution

18 Sep 86 20:21:00 EDT

that effect, from an SDI spokesman referring to a recent test.

Let's take this with a grain of salt. I have seen a large system (over 100,000 lines of high-level language) "work the first time". By this I mean that in the first live test of the system, it performed as designed with no errors. That software had been designed and programmed by a small, close-knit group of experienced real-time programmers, and had been extensively tested at the module level with drivers and stubs, and also at the system level using a very realistic simulation. (Also bear in mind that the first live test of *any* system is likely to be quite conservative in its objectives; it's likely that only a small fraction of all possible paths through the code will actually be exercised.) Furthermore, the 100K lines of code that made it to the first live test were by no means the original, first-cut 100K lines written (although a gratifyingly large percentage of them were, thanks to good design practices.)

If the SDI test were a similar situation -- well-designed, thoroughly pre-tested software that worked well on its initial, conservative live test -- then it's at least plausible. If, on the other hand, the spokesman actually meant "we coded up 1,000,000 lines and then tried them and they all worked" -- then I'd have to see some proof (in fact, a *lot* of proof) before I'd believe it.

A million lines of code works the first time?

Dave Benson <benson%wsu.csnet@CSNET-RELAY.ARPA>
Tue, 16 Sep 86 16:56:14 pdt

|Heard on NPR's "All Things Considered" yesterday evening:
|An Air Force Lt. Col., speaking about a kinetic energy weapons
|test earlier this week, which apparently went better than expected
|in several respects. If this isn't an exact quote (I heard it
|twice, but didn't write it down at the time), it's real close:

|"We wrote about a million lines of new computer code, and tested
|them all for the first time, and they all worked perfectly."

Hoo boy! I would appreciate any and all leads by which I might track this to some reliable source. Thank you, David B. Benson, Computer Science Department, Washington State University, Pullman, WA 99164-1210. csnet: benson@wsu

---

I found one! (A critical real-time application worked the first time)

<LIN@XX.LCS.MIT.EDU>
Wed, 17 Sep 1986 12:44 EDT

From: Dave Benson

Re: Massive UNIX breakins at Stanford

"Scott E. Preece" <preece%ccvaxa@GSWD-VMS.ARPA>
Thu, 18 Sep 86 09:12:59 cdt

> From: reid@decwrl.DEC.COM (Brian Reid) The machine on which the initial
> breakin occurred was one that I didn't even know existed, and over
> which no CS department person had any control at all. The issue here is
> that a small leak on some inconsequential machine in the dark corners
> of campus was allowed to spread to other machines because of the
> networking code. Security is quite good on CSD and EE machines, because
> they are run by folks who understand security. But, as this episode
> showed, that wasn't quite good enough.

No you're still blaming the networking code for something it's not supposed to do. The fault lies in allowing an uncontrolled machine to have full access to the network. The NCSC approach to networking has been just that: you can't certify networking code as secure, you can only certify a network of machines AS A SINGLE SYSTEM. That's pretty much the approach of the Berkeley code, with some grafted on protections because there are real-world situations where you have to have some less-controlled machines with restricted access. The addition of NFS makes the single-system model even more necessary.

scott preece, gould/csd - urbana, uucp: ihnp4!uiucdcs!ccvaxa!preece

---

Re: Protection of personal information

<Andy_Mondore%RPI-MTS.Mailnet@MIT-MULTICS.ARPA>
Fri, 19 Sep 86 10:00:10 EDT

David Chase wrote in Risks 3.58 that at his university, students were required to give a lot of personal information on a form before they could sign up for on-campus job placement interviews and that by signing this form, they authorized the university to release their transcripts to potential employers. He also complained about the use of the social security number as the student number.

Here at RPI, I think the only form you are required to fill out before getting on-campus interviews is a resume form. I work in the Registrar's office and we release a transcript only if we have received a signed statement from the student authorizing release of the transcript to a specific person or company. As far as I know, we don't accept "blanket" releases.

As for the use of social security numbers as student numbers -- we also use social security numbers for this purpose. One of the reasons we do this is that if you are receiving financial aid, we must verify your attendance every semester to the agency supplying the aid. Very often, this verification is in the form of a computer-generated list or tape from the agency and the only way to cross-reference their list to our file is via the social security number. It is usually difficult to do a computer-match on name because of differences in how the names might be formatted. There is the same problem when a student has an on-campus job -- the payroll office needs to verify that the student is registered and they need the social security number for tax purposes, so they prefer to use it as their primary means of identifying the student (or any other employee).

In terms of requiring you to give us your social security number, federal law prohibits us from requiring you to give it to us except for tax or social security purposes. However, the law has also been interpreted to mean that we also have the option of not servicing you if you refuse to give it. (I don't think that has ever happened here, however.)

For the final word on what can and cannot be done with personal information, I suggest you check the Family Rights to Privacy Act (popularly known as the Buckley Amendment).

Protection of personal information

<LIN@XX.LCS.MIT.EDU>
Thu, 18 Sep 1986 21:26 EDT

My understanding is that use of one's SS number must be authorized by law. There are times when others ask, but you are not required to give it to them.

Under those circumstances, I don't believe it it is illegal to give a fake SSN. The way to protect yourself is to give your real SSN, except for a small error that you can later blame on an entry error.

Announcement of Berkeley Conference on the SDI

Eric Roberts <roberts@src.DEC.COM>
Thu, 18 Sep 86 13:25:05 pdt

The Dave Redell/Hugh DeWitt panel (Saturday morning) should be of special interest to RISKS readers and the rest of the program of general interest.

STAR WARS AND NATIONAL SECURITY

A Conference on the Strategic Defense Initiative
October 9-11, 1986, University of California, Berkeley

Thursday Evening, 8:00-10:30, Wheeler Auditorium

Opening Debate: "Technical Feasibility and Strategic Policy Implications of the SDI"

Andrew Sessler (moderator), Former Director of Lawrence Berkeley Laboratory; Member of American Physical Society Panel on Directed Energy Weapons.
Lowell Wood, leader of "O Division," Lawrence Livermore National Laboratories.
Richard Garwin, IBM Research Fellow; Adjunct Professor of Physics, Columbia University; Adjunct Research Fellow, Center for Science and International Affairs, Kennedy School of Government, Harvard University.
Colin Gray, President, National Institute for Public Policy; Member of the President's General Advisory Committee on Arms Control and Disarmament.
John Holdren, Professor of Energy and Resources, UC Berkeley; Chairman, U.S. Pugwash Committee; Former Chairman, Federation of American Scientists.

Friday Morning, 9:00-11:00, Sibley Auditorium

Legislative Hearing: "Keeping California Competitive in R&D: The Impacts of Increased Military Spending, the SDI, and Federal Tax Reform" (This event will be co-sponsored by the California Assembly Committee on Economic Development and New Technologies.)

Glenn Pascall, Senior Research Fellow, Graduate School of Public Affairs, University of Washington; President, Columbia Group Inc.
Jay Stowsky, Research Economist, Berkeley Roundtable on the International Economy, UC Berkeley.
Ted Williams, Chief Executive Officer, Bell Laboratories [invited].
Robert Noyce, Vice-Chairman of the Board, Intel [invited].
Ralph Thompson, Senior Vice-President for Public Affairs, American Electronics Association.
John Holdren, Professor of Energy and Resources, UC Berkeley; Chairman, U.S. Pugwash Committee; Former Chairman, Federation of American Scientists.

Documentary Film: "Star Wars: A Search for Security," produced by Ian Thiermann for PSR, 11:30-12:00 and 2:00-2:30, Room 4, Dwinell Hall.

Friday Afternoon, 3:00-5:00, Wheeler Auditorium

Panel Discussion: "The Effects of SDI on Universities"
Marvin Goldberger (moderator), President, Caltech.
Vera Kistiakowsky, Professor of Physics, MIT.
John Holdren, Professor of Energy and Resources, UC Berkeley; Chairman, U.S. Pugwash Committee; Former Chairman, Federation of American Scientists.
Clark Thompson, Professor of Computer Science, University of Minnesota.
Danny Cohen, Director, Systems Division, Information Sciences Institute, University of Southern California; Chairman, SDIO Committee on Computing in Support of Battle Management.

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 61

Sunday, 21 September 1986

Contents

- Computers and Ethics
  Robert Reed
- Autonomous weapons
  Wayne Throop
- Simulation risk
  Rob Horn
- Viking software
  James Tomayko
- Risks of passwords on networks
  Bruce
- More on digital jets; Sanity checks
  Eugene Miya
- Info on RISKS (comp.risks)

Computers and Ethics

<bobr%zeus.tek.csnet@CSNET-RELAY.ARPA>
19 Sep 86 13:36:54 PDT (Fri)

In RISKS-3.54 Mark S. Day writes:
> ...people will read mail and data files stored on a timesharing system, even
> though it's unacceptable to rifle through people's desks. [...]

It occurs to me that each of these suggested mechanisms can be interpreted in different ways which may provide new insights into the problem.

Novelty. Social conditioning aside, the thrill of adventure in a new environment leads many people to explore the system in a quest for new understanding about it. It is perhaps easier to lay the moral questions aside when caught in the fervor of covering new ground. In fact the thrill is enhanced by doing something slightly larcenous.

Distance. Certainly the distance between people is greater, but the

distance between private pathways is shorter.
Psychologically, I feel closer to your portion of the file
system than I do to the contents of your desk drawers.
Especially if working in an environment where limited
sharing of files is part of the norm, the sense of
territorial lines is less distinct within such an electronic
medium

There is a third aspect which is related to the thrill factor, and that is the threat of being caught. If I am found in your office with my hand in your desk, the evidence is pretty compelling and not easy to hide. Within a computer system, we are all little "virtual people", moving silently around the directory tree, and so much less likely to arouse suspicions, so even when ethical considerations are present, the concern about getting caught is lessened by the nature of the medium.

Robert Reed, Tektronix CAE Systems Division, bobr@zeus.TEK

Autonomous weapons

<rti-sel!dg_rtp!throopw%mcnc.csnet@CSNET-RELAY.ARPA>
Fri, 19 Sep 86 16:46:17 edt

> eugene@AMES-NAS.ARPA (Eugene Miya)
> Most recently, another poster brought up the issue of autonmous weapons.

It is worth pointing out that we are *currently* using autonomous weapons and they are *not* smart enough to distinguish signs of surrender. Give up? I'm talking about, for example, hand grenades or landmines. These are autonomous (after being thrown or burried) and their mission (guided by a particularly simple "computer") is to saturate their environment with shrapnel after a suitable delay. Bombs with proximity fuses, self-guided missiles, and so on, where there is "intelligence" in the weapon and a significant time delay between the decision to deploy and the weapon's effective discharge can all be considered cases of "autonomous weapons". We are (in this view) simply trying to make the beasties smarter, so that they eventually *will* be able to recognize signs of surrender or cease-fire or other cases of cessation of hostilities. (Picture land-mines getting up and "trooping" back to an armory after the war is over...)

Perhaps this is more appropos to one of the "arms" lists, but I think it is worth noting that we are allowing some *very* simple "computers" to be in charge of some *very* powerful weapons right now. It is an interesting question to ask if we really *want* to make the weapons smarter. But I don't think it is a question of whether to use autonomous weapons at all... we're already using them.

Wayne Throop <the-known-world>!mcnc!rti-sel!dg_rtp!throopw

Simulation risk

Rob Horn <harvard!wanginst!infinet!rhorn@seismo.CSS.GOV>
Sat, 20 Sep 86 16:11:42 edt

One kind of risk that I have not seen discussed here is the problems posed by using computer simulation models that are not adequate. In particular I am refering to situations where due to either insufficient computer resources, or insufficient mathematical analysis, the really accurate model results are not available. Usually more primitive, inaccurate model results are available and being used by the ideologues on both sides of an issue. This places the responsible scientists and engineers in a difficult situation. How do you say ``I don't know yet'' and how do you deal with making recommendations in the absence of adequate information.

I can think of two such situations that have major public decision-making impact.

The first is the ``nuclear winter'' situation. I remember many years ago reading the sensitivity analysis of the one-dimensional and two-dimensional climate models to solar input. They were hyper-sensitive, with variations on the order of measurement error causing massive climate change. It was not until recently (1982) that the vectorized Climate Model was analyzed and shown to be reasonably well behaved. And even it has some contentious approximations. This model requires 15 hours on a CRAY-1 to analyze one situation for one season.

When the nuclear winter stories came out I had my doubts. Where did these people find a solid month (12 seasons x 4(?) test cases) of CRAY time? Had they used one of the hyper-sensitive 1 or 2-dimensional models. What would the accurate models find? And how should I respond when I knew that it would probably be a year or more before that much CRAY time and post-simulation analysis could be finished? (Fortunately I only had to handle party conversation with people who knew that I had worked in that field.)

The same kind of problem occured in the ozone layer issues during the mid 70's. The more accurate model had two extremely severe problems: 1) it was unconditionally unstable when phrased as a finite difference problem or exceedingly temperamental when phrased as an implicit differencing problem. 2) It involved solving extremely stiff differential equations. In this case the official answer given was ``we don't know. It will take several years of mathematical research effort to make this problem tractable. The real answer is anyone's guess. The published model answers are meaningless.'' A truthful answer but of little value to decision makers. (There was a brute force throw-computers-at-it solution. Estimated run-time on a CRAY was about 1,000 years per simulated year. Easier to wait and see what happened.)

How often are we placed in a situation where the inaccurate computer simulation is available, but the accurate simulation unavailable?
What is an appropriate way to deal with this problem?

Rob Horn
UUCP: ...{decvax, seismo!harvard}!wanginst!infinet!rhorn
Snail: Infinet, 40 High St., North Andover, MA

Viking software

<James.Tomayko@sei.cmu.edu>
Sunday, 21 September 1986 09:25:25 EDT

The Viking software load for the lander was 18K words stored on plated wire memory. The Martin Marietta team decided to use a 'software first' approach to the development of the flight load. This meant a careful examination of the requirements, a serious memory estimate, and then commitment by the project team to stay within that memory estimate. The software was developed on an emulator that used microcoded instructions to simulate the as-yet-unpurchased computer. Sources for this are a Rome Air Development Center report that studied software development, later summarized in a book by Robert L. Glass. The Viking software documents for the orbiter, developed by JPL, are so good I use them as examples of tracability in my current software engineering courses.

Risks of passwords on networks

<BRUCE%UC780.BITNET@WISCVM.WISC.EDU>
20 SEP 86 14:57-EST

A few thoughts about networks which ask for passwords to send files. Take a computer network with three computers. Call them computer A, B, and C. Computer user on A wants to send a file to their account on C through computer B. No problem, we invoke the command to send files, supply it with a password (and maybe a username at computer C) and off the files go. But, on computer B, there is a "smart" systems programmer who monitors all network traffic through his/her node. How interesting... A file copy operation with a user name/password attached.

The point? Just a password is not a good solution. Maybe one would need to encrypt the packets through the network (so that intermediate nodes couldn't read them).

Bruce

More on digital jets; Sanity checks

Eugene Miya <eugene@AMES-NAS.ARPA>
Sat, 20 Sep 86 11:40:44 pdt

Talk about timing:

In the latest copy of IEEE Spectrum (why didn't anyone else post this?)

%A Cary R. Spitzer
%Z NASA LaRC, Hampton, VA
%T All-digital Jets are Taking Off

%J IEEE Spectrum
%V 23
%N 9
%D September 1986
%P 51-56
%X Covers F-14D, F-16[CD], A-3[012]) airbus, 7J7, MD-11, and
other 1st and emerging 2nd generation digital systems.
Has good basic references.

Added note. I will be contacting some old Viking friends for a further
detailed description and references as requested (probably next Tu. or We
when they come up here).

On Sanity checks:

I had a similar incident in a Silicon Valley Mexician restaurant
which I reported in a early RISK to the pocket book. This issue
has appeared other news groups like mod.comp-soc on the USENET.
I offer the following reference:

%A Jon L. Bentley
%Z ATT BL (research!)
%T The Envelope is Back
%J Communications of the ACM
%S Programming Pearls
%V 29
%N 3
%D March 1986
%P 176-182
%K rules of thumb, cost, orders of magnitude, quick calculations,
Litle's Law
%X JLB's principles include:
Familiarity with numbers
Willingness to experiment [actively, discussing this one with Denning]
Discipline in checking answers
Mathematics, when you need it
He also gives the "Back of the Envelope" column in the
American Journal of Physics as good reading.

I am reminded of a quote by Eric Shipton, an early English Mt. Everest veteran
who died recently: (paraphased) Never go on an expedition which you can't
plan on the back of an envelope. I know this is how spaceflight is
frequently done.

--eugene miya
NASA ARC

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 62

Monday, 22 September 1986

Contents

- Massive UNIX breakins at Stanford
  - Jerry Saltzer
  - Rob Austein
  - Andy Freeman
  - Scott Preece
- F-16 Software
  - Henry Spencer
- 1,000,000 lines of correct code?
  - Stephen Schaefer
- Info on RISKS (comp.risks)

Massive UNIX breakins at Stanford

Jerome H. Saltzer <Saltzer@ATHENA.MIT.EDU>
Mon, 22 Sep 86 11:04:16 EDT

In RISKS-3.58, Dave Curry gently chastises Brian Reid:

> . . . you asked for it. . . Berkeley networking had nothing to
> do with your intruder getting root on your system, that was due purely
> to neglect. Granted, once you're a super-user, the Berkeley networking
> scheme enables you to invade many, many accounts on many, many machines.

And in RISKS-3.59, Scott Preece picks up the same theme, suggesting that Stanford failed by not looking at the problem as one of network security, and, in the light of use of Berkeley software, not enforcing a no-attachment rule for machines that don't batten down the hatches.

These two technically- and policy-based responses might be more tenable if the problem had occurred at a military base. But a university is a different environment, and those differences shed some light on environments that will soon begin to emerge in typical commercial and networked home computing settings. And even on military bases.

There are two characteristics of the Stanford situation that
RISK-observers should keep in mind:

1. Choice of operating system software is made on many factors, not just the quality of the network security features. A university has a lot of reasons for choosing BSD 4.2. Having made that choice, the Berkeley network code, complete with its casual approach to network security, usually follows because the cost of changing it is high and, as Brian noted, its convenience is also high.

2. It is the nature of a university to allow individuals to do their own thing. So insisting that every machine attached to a network must run a certifably secure-from-penetration configuration is counter-strategic. And on a campus where there may be 2000 privately administered Sun III's, MicroVAX-II's, and PC RT's all running BSD 4.2, it is so impractical as to be amusing to hear it proposed. Even the military sites are going to discover soon that configuration control achieved by physical control of every network host is harder than it looks in a world of engineering workstations.

Brian's comments are very thoughtful and thought-provoking. He describes expected responses of human beings to typical current-day operating system designs. The observations he makes can't be dismissed so easily.

Jerry Saltzer

Massive UNIX breakins at Stanford

Rob Austein <SRA@XX.LCS.MIT.EDU>
Mon, 22 Sep 1986 23:03 EDT

I have to take issue with Scott Preece's statement that "the fault lies in allowing an uncontrolled machine to have full access to the network". This may be a valid approach on a small isolated network or in the military, but it fails horribly in the world that the rest of us have to live in. For example, take a person (me) who is (theoreticly) responsible for what passes for security on up to half a dozen mainframes at MIT (exact number varies). Does he have any control over what machines are put onto the network even across the street on the MIT main campus? Hollow laugh. Let alone machines at Berkeley or (to use our favorite local example) the Banana Junior 6000s belonging to high school students in Sunnyvale, California.

As computer networks come into wider use in the private sector, this problem will get worse, not better. I'm waiting to see when AT&T starts offering a long haul packet switched network as common carrier.

Rule of thumb: The net is intrinsicly insecure. There's just too much cable out there to police it all. How much knowledge does it take to tap into an ethernet? How much money? I'd imagine that anybody with a BS from a good technical school could do it in a week or so for under $5000 if she set her mind to it.

As for NFS... you are arguing my case for me. The NFS approach to security seems bankrupt for just this reason. Same conceptual bug, NFS simply agravates it by making heavier use of the trusted net assumption.

Elsewhere in this same issue of RISKS there was some discussion about the dangers of transporting passwords over the net (by somebody other than Scott, I forget who). Right. It's a problem, but it needn't be. Passwords can be tranmitted via public key encryption or some other means. The fact that most passwords are currently transmitted in plaintext is an implementation problem, not a fundamental design issue.

A final comment and I'll shut up. With all this talk about security it is important to keep in mind the adage "if it ain't broken, don't fix it". Case in point. We've been running ITS (which has to be one of the -least- secure operating systems ever written) for something like two decades now. We have surprisingly few problems with breakins on ITS. Seems that leaving out all the security code made it a very boring proposition to break in, so almost nobody bothers (either that or they are all scared off when they realize that the "command processor" is an assembly language debugger ... can't imagine why). Worth thinking about. The price paid for security may not be obvious.

--Rob Austein <SRA@XX.LCS.MIT.EDU>

Massive UNIX breakins at Stanford

Andy Freeman <ANDY@Sushi.Stanford.EDU>
Mon 22 Sep 86 11:07:04-PDT

Scott E. Preece <preece%ccvaxa@GSWD-VMS.ARPA> writes in RISKS-3.60:

> reid@decwrl.DEC.COM (Brian Reid) writes:
> The issue here is that a small leak on some [unknown] inconsequential machine in the dark corners of campus was allowed to spread to other machines because of the networking code.
>
> No, you're still blaming the networking code for something it's not supposed to do. The fault lies in allowing an uncontrolled machine to have full access to the network. The NCSC approach to networking has been just that: you can't certify networking code as secure, you can only certify a network of machines AS A SINGLE SYSTEM. That's pretty much the approach of the Berkeley code, with some grafted on protections because there are real-world situations where you have to have some less-controlled machines with restricted access. The addition of NFS makes the single-system model even more necessary.

Then NCSC certification means nothing in many (most?) situations. A lot of networks cross adminstrative boundaries. (The exceptions are small companies and military installations.) Even in those that seemingly don't, phone access is often necessary.

Network access should be as secure as phone access. Exceptions may choose to disable this protection but many of us won't. (If Brian didn't know about the insecure machine, it wouldn't have had a valid password to access his machine. He'd also have been able to choose what kind of access it had.) The only additional problem that networks pose is the ability to physically disrupt other's communication.

-andy [There is some redundancy in these contributions, but each makes some novel points. It is better for you to read selectively than for me to edit. PGN]

Massive UNIX breakins at Stanford (RISKS-3.60)

"Scott E. Preece" <preece%mycroft@GSWD-VMS.ARPA>
22 Sep 1986 16:24-CST

Andy Freeman writes [in response to my promoting the view of a network as a single system]:

> Then NCSC certification means nothing in many (most?) situations.

Well, most sites are NOT required to have certified systems (yet?). If they were, they wouldn't be allowed to have non-complying systems. The view as a single system makes the requirements of the security model feasible. You can't have anything in the network that isn't part of your trusted computing base. This seems to be an essential assumption. If you can't trust the code running on another machine on your ethernet, then you can't believe that it is the machine it says it is, which violates the most basic principles of the NCSC model. (IMMEDIATE DISCLAIMER: I am not part of the group working on secure operating systems at Gould; my knowledge of the area is superficial, but I think it's also correct.)

[NOTE: The word "NOT" in the first line of this paragraph was interpolated by PGN as the presumed intended meaning.]

Network access should be as secure as phone access. Exceptions may choose to disable this protection but many of us won't. (If Brian didn't know about the insecure machine, it wouldn't have had a valid password to access his machine. He'd also have been able to choose what kind of access it had.) The only additional problem that networks pose is the ability to physically disrupt other's communication.

Absolutely, network access should be as secure as phone access, IF YOU CHOOSE TO WORK IN THAT MODE. Our links to the outside world are as tightly restricted as our dialins. The Berkeley networking software is set up to support a much more integrated

kind of network, where the network is treated as a single system.
For our development environment that is much more effective.
You should never allow that kind of access to a machine you don't
control. Never. My interpretation of the original note was that
the author's net contained machines with trusted-host access
which should not have had such access; I contend that that
represents NOT a failing of the software, but a failing of the
administration of the network.

scott preece
gould/csd - urbana, uucp: ihnp4!uiucdcs!ccvaxa!preece

F-16 Software

<ihnp4!utzoo!henry@ucbvax.Berkeley.EDU>
Mon, 22 Sep 86 18:07:11 PDT

Doug Wade notes:

> My comment to this, is what if a 8G limit had been programmed into
> the plane (if it had been fly-by-wire)...

My first reaction on this was that military aircraft, at least front-line combat types, obviously need a way to override such restrictions in crises, but civilian aircraft shouldn't. Then I remembered the case of the 727 that rolled out of control into a dive a few years ago. The crew finally managed to reduce speed enough to regain control by dropping the landing gear. The plane was at transonic speed at the time -- there was some speculation, later disproven, that it might actually have gone slightly supersonic -- and was undoubtedly far above the official red-line maximum airspeed for the landing gear. It would seem that even airliners might need overrides.

Henry Spencer @ U of Toronto Zoology
{allegra,ihnp4,decvax,pyramid}!utzoo!henry

1,000,000 lines of correct code?

Stephen Schaefer <schaefer%research.bgsu.edu@CSNET-RELAY.ARPA>
Mon, 22 Sep 86 19:15:31 edt

The Plain Dealer (Cleveland), Tuesday, September 16, 1986
Excerpted without permission.

"Protecting the secrets of success"

Dayton(AP) - [Most of article dealing with foreign contractors omitted] [Col. Thomas D.] Fiorino also said a Sept. 5 experiment using two satellites that measured the plume of a rocket exhaust in space and then collided was a success. Some critics, noting the experiment took 1 million lines of computer code, said a full SDI system would take tens or hundreds of millions.

  Fiorino said there was a computer on board that processed 2
 billion operations a second, about four times faster than current
 "supercomputers."
  "It did not represent our full technological potential," he
 said, pointing out that it did not use very high speed integrated
 circuits still under development.

On the one hand, I am incredulous, but on the other, I'd be utterly horrified to find them directing misinformation to the small number of people knowledgeable enough to understand. I hope this ruggedized, portable, Cray class machine is commercially available in a couple years. Failing that, I hope the reporter was simply "innumerate" and heard "billion" for "million" somewhere.

I must repeat the quote of Mark Twain by the original poster:
"Interesting if true - and interesting anyway."

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 63

Wednesday, 24 September 1986

Contents

- NOTROJ (a Trojan Horse)
  James H. Coombs via Martin Minow
- Massive UNIX breakins at Stanford
  Scott Preece [two more messages!]
- Info on RISKS (comp.risks)

<minow%regent.DEC@decwrl.DEC.COM>
23-Sep-1986 1644

(Martin Minow, DECtalk Engineering ML3-1/U47 223-9922)
To: risks@csl.sri.com
Subject: NOTROJ (a Trojan Horse)

Found on a local bboard:

Date: Sat, 20 Sep 86 04:17:50 EDT
From: "James H. Coombs" <JAZBO%BROWNVM.BITNET@WISCVM.WISC.EDU>
Subject: NOTROJ--it IS a trojan

Distribute far and wide!
(C)Nobodaddy, 1986

A Story of a Trojan Horse
With Some Suggestions for Dismounting Gracefully

by
James H. Coombs

NOTROJ.COM is a TROJAN HORSE (comes in NOTROJ.ARC--for now).

I first became aware of NOTROJ when a member of The BOSS BBS community reported his belief that the program destroyed the directory of his hard disk. After two days of restoring his files, he concluded:

> This Trojan was written by a real Pro---he knows his ASM and uses it as a weapon---not a tool. From lokkin' at the job he did on me, I tendto doubt that I would have found the bomb has I been smart enough to look. ---PLEASE!!!!! Spread the word 'bout this one. It's a Killer!

In the next couple of days, I saw a similar note on the Boston Computer Society bulletin board. This victim rather pathetically credits NOTROJ with a "valiant" attempt at saving his data.

> The program in question is a time-bomb (about 10 minutes) and works by the "SOFTGUARD UNFORMAT" method of attack. I'm not sure what it did, or how it did it, or even how I could have recovered the disk but the NOTROJ program I had in the background alerted me to the fact, and tried a valiant attempt to shut down the hard disk. To no avail, though.

Since my hard disk was becoming fragmented anyway, I decided to test NOTROJ. Everything looked pretty reasonable from the start; in fact, the program looks like a very useful tool (although I'm not in love with the interface). One loads NOTROJ resident and then accesses the options menu through Alt-N. The menu contains about fifteen items, some of them annotated "DANGER", e.g., "Format track (DANGER!)". For each parameter, the user can select one of four responses: Proceed, Timeout, Reboot, or Bad Command. The menu also provides a fifth option--"Pause&Display"--which provides the user with full information on the activity that the currently active program is trying to perform and prompts for one of the four primary actions, e.g, Proceed.

I selected "Pause&Display" for all of the DANGERous parameters. Everything worked fine, although I found that iteratively selecting "Timeout" in response to the "Write sectors" interrupt hung up the machine. I fooled around with a number of commands and finally reproduced the disk crash. At the time, I was running the DOS ERASE command (I had been suspicious of that one for quite some time anyway). I don't have the full message that the program displayed, but I did write down this much "Softguard-style low-level disk format." (Keep those words in mind.)

In spite of the fact that I had prepared for a disk crash, it took me at least an hour to get running again. When I booted the machine, I was thrown into BASIC and could not get back to the system. I put a DOS diskette in, and got an invalid drive error message when I tried to access the hard disk. Here is the recovery procedure for this and most disk crashes:

1) Insert DOS system disk in drive A.
2) Reboot the machine.
3) Run FDISK and install a DOS partition on the hard disk.
4) Format the hard disk with the '/S' option.
5) Restore files from the most recent full-disk Bernoulli or tape

backup.
6) Restore files modified since the most recent full-disk Bernoulli
or tape backup.

Once I got a minimal system running, I decided to reproduce the crash to ensure that this was not some quirk of bad programming. What, ho! I got bored playing around with COPY and ERASE and a few other programs. I waited for a while, read a magazine--no signs of a simple timing technique. I began to think that NOTROJ might be more incompetent than vicious. Something about the documentation made it seem unlikely that the author was a criminal. It occurred to me, however, that the author might have had some time to waste on this program. Does he, perhaps, check to see how full the hard disk is? It would be reasonable to evade detection immediately after a bomb by making it impossible to reproduce the crash. In addition, it would be much more painful for people if they have restored all of their files or gradually rebuilt their hard disks before they discover that this is a trojan horse. So, I restored all of my files.

This time, Norton's NU command turned out to be the great blackguard that was trying to format my disk (according to NOTROJ--although it was only reading the FAT). So, I restored my hard disk. All of the while, however, I had the nagging feeling that the documentation did not reflect the personality of someone vicious. When I got running again, I took a look into NOTROJ.COM. Nowhere could I find the words from the message "Softguard-style low-level disk format." That convinced me. I have concealed passwords on mainframes by assembling strings dynamically instead of storing them statically. Our trojanette must have used the same technique so that no one would spot the suspicious messages. I had counted on being able to get them directly from the program so that I would not have to take the time to write the whole message down while my system was being operated on. I do recall NOTROJ patting itself on the back, however, for preventing "further damage."

As I think back on it, the documentation contains something of a rant against copy-protection schemes, including Softguard. In addition, I had always been troubled by the fact that the name NOTROJ is an acrostic for TROJAN and also an assertion that the program is not itself a trojan. The documentation is also very badly written. One has to experiment to make sense of it, although that is nothing new in software documentation. Also, the style is something of a pidgin English, which seems consistent with the fact that the author has an Oriental name (Ng, or is that for "no good"?). Well, since the author's name and address are listed in the documentation, I decided to give him a call. Mirabile dictu! It's a real name, and I got a real number--I just didn't get an answer, even at 2 a.m. It doesn't make much difference anyway, there's nothing that he can say to convince me that he had legitimate reasons for concealing error messages and that his program is not a trojan horse. There is also the possibility that the person listed as author has nothing to do with the program. Could the pidgin style of the documentation be the work of a clever linguist--an acrostic fan--a sick person who considers himself to be the bozo that Sherlock Holmes was always after? Who knows? I have to write a book. No time to play with these fools.

So, be careful. Note that sysops don't have the time to test every program extensively. If a program like NOTROJ requires that a disk be more than 70% full, for example, a lot of people may never have any problems with it. What else can we do? Does someone want to try to prosecute the author of NOTROJ? And how do we keep ourselves from becoming paranoid about new noncommerical software?

Eventually, I think it will all shake out just fine. Those of us who are prepared for problems provide others with the testing and filtering. Junk like NOTROJ just does not make it into my community. Actually, I find mediocre software much more of a problem. I have spent a lot of time and money sorting through megabytes of chaff to find but a few grains of wheat. I would like to see us find some way to constrict the growth of chaff and worms both. If we can't do this, many of us may have to switch to commercial software.

--Jim

Replies may be made to:
BITNET: JAZBO@BROWNVM
BBS: The BOSS, BCS, Hal's, et passim
BIX: jcoombs

Massive UNIX breakins at Stanford

"Scott E. Preece" <preece%ccvaxa@GSWD-VMS.ARPA>
Tue, 23 Sep 86 09:16:21 cdt

[This was an addendum to Scott's contribution to RISKS-3.61. PGN]

I went back and reviewed Brian Reid's initial posting and found myself more in agreement than disagreement. I agree that the Berkeley approach offers the unwary added opportunities to shoot themselves in the foot and that local administrators should be as careful of .rhosts files as they are of files that are setuid root; they should be purged or justified regularly.

I also agree that it should be possible for the system administrator to turn off the .rhosts capability entirely, which currently can only be done in the source code and that it would be a good idea to support password checks (as a configuration option) on rcp and all the other remote services.

scott preece, gould/csd - urbana, uucp: ihnp4!uiucdcs!ccvaxa!preece

Re: Massive UNIX breakins at Stanford

"Scott E. Preece" <preece%ccvaxa@GSWD-VMS.ARPA>
Tue, 23 Sep 86 08:41:29 cdt

> From: Rob Austein <SRA@XX.LCS.MIT.EDU>

> I have to take issue with Scott Preece's statement that "the fault lies
> in allowing an uncontrolled machine to have full access to the network"...

I stand by what I said, with the important proviso that you notice the word "full" in the quote. I took the description in the initial note to mean that the network granted trusted access to all machines on the net. The Berkeley networking code allows the system administrator for each machine to specify what other hosts on the network are to be treated as trusted and which are not. The original posting spoke of people on another machine masquerading as different users on other machines; that is only possible if the (untrustworthy) machine is in your hosts.equiv file, so that UIDs are equivalenced for connections from that machine. If you allow trusted access to a machine you don't control, you get what you deserve.

Also note that by "the network" I was speaking only of machines intimately connected by ethernet or other networking using the Berkeley networking code, not UUCP or telephone connections to which normal login and password checks apply.

The description in the original note STILL sounds to me like failure of administration rather than failure of the networking code.

scott preece

[OK. Enough on that. The deeper issue is that most operating systems are so deeply flawed that you are ALWAYS at risk. Some tentative reports of Trojan horses discovered in RACF/ACF2 systems in Europe are awaiting details and submission to RISKS. But their existence should come as no surprise. Any use of such a system in a hostile environment could be considered a failure of administration. But it is also a shortcoming of the system itself... PGN]

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 64

Wednesday, 24 September 1986

Contents

- Sane sanity checks / risking public discussion
  Jim Purtilo
- More (Maybe Too Much) On More Faults
  Ken Dymond
- Re: Protection of personal information
  Correction from David Chase
- Towards an effective definition of "autonomous" weapons
  Herb Lin
  Clifford Johnson [twice each]
- Info on RISKS (comp.risks)

Sane sanity checks / risking public discussion

Jim Purtilo <purtilo@brillig.umd.edu>
Tue, 23 Sep 86 12:54:10 EDT

[Regarding ``sanity checks'']

Let us remember that there are sane ``sanity checks'' as well as the other kind. About 8 years ago while a grad student at an Ohio university that probably ought to remain unnamed, I learned of the following follies:

The campus had long been doing class registration and scheduling via computer, but the registrar insisted on a ``sanity check'' in the form of hard copy. Once each term, a dozen guys in overalls would spend the day hauling a room full of paper boxes over from the CS center, representing a paper copy of each document that had anything to do with the registration process. [I first took exception to this because their whole argument in favor of "computerizing" was based on reduced costs, but I guess that should be hashed out in NET.TREE-EATERS.]

No one in that registrar's office was at all interested in wading through all that paper. Not even a little bit.

One fine day, the Burroughs people came through with a little upgrade to the processor used by campus administration. And some "unused status bits" happened to float the other way.

This was right before the preregistration documents were run, and dutifully about 12,000 students preregistration requests were scheduled and mailed back to them. All of them were signed up "PASS/FAIL". This was meticulously recorded on all those trees stored in the back room, but no one wanted to look.

I suppose a moral would be ``if you include sanity checks, make sure a sane person would be interested in looking at them.''

[Regarding break-ins at Stanford]

A lot of the discussion seems to revolve about ``hey, Brian, you got what you asked for'' (no matter how kindly it is phrased). Without making further editorial either way, I'd like to make sure that Brian is commended for sharing the experience. Sure would be a shame if ``coming clean'' about a bad situation will be viewed as itself constituting a risk...

[I am delighted to see this comment. Thanks, Brian! PGN]

More (Maybe Too Much) On More Faults

"DYMOND, KEN" <dymond@nbs-vms.ARPA>
23 Sep 86 09:18:00 EDT

The intuitive sense made by Dave Benson's argument in RISKS 3.50, that

>We need to understand that the more faults found at any stage to
>engineering software the less confidence one has in the final product.
>The more faults found, the higher the likelihood that faults remain.

seems to invite a search for confirming data because there are also counter-intuitive possibilities. For example there is the notion that the earlier in the life cycle errors are detected, the cheaper to remedy them. There is a premium on finding faults early. And the further notion that with tools for writing requirements in some kind of formal language that can be checked for syntactic and semantic completeness and consistency, it's possible to detect at least some errors at requirements stage that may not have been caught till later. So SE projects using these and similar methods for other stages in the life cycle would tend to show more errors earlier. Would the products from these projects be therefore less reliable than others made with, say, more traditional, less careful, design and programming practice ?

Dave makes the further argument in RISKS 3.57:

>Certain models of software failure place increased "reliability" on
>software which has been exercised for long periods without fault. [...]

The models of software reliability exist to order our thinking about reliability and to help predict behavior of software systems based on observation of failure up to the current time. The models that show failures clustered early in time and then tapering off later do indeed model an intuition but maybe not the one that more faults mean yet more faults. Hence the need for data. I suspect that the reality as shown by data, if it exists, would be more complex than intuition allows. More errors discovered so far may just mean better software engineering methods. As far as other engineering fields, the failure vs time curve in manufactured products is often taken to be tub-shaped, not exponentially decaying. So more failures are expected at the beginning and near the end of the useful life of a "hard" engineered product. Of course, "an unending sequence of irremediable faults" should be the kiss of death for any product, whether from hard engineering or soft. But the trick is in knowing that the sequence is unending. The B-17, I seem to remember reading, had a rather rocky development road in the 1930s, yet was not abandoned. Was it just that the aeronautical engineers at Boeing then had in mind some limit on the number of faults and that this limit was not exceeded? It might be easy to say in hindsight. On the other hand, sometimes foresight, in terms of spotting a poor design at the outset, makes a difference, as in the only Chernobyl-type power reactor outside the Soviet block. It was bought by Finland (perhaps this is what "Finlandization" means ?). However the Finns also bought a containment building from Westinghouse.

Ken Dymond

Re: Protection of personal information

David Chase <rbbb@rice.edu>
Tue, 23 Sep 86 08:56:18 EDT

[The two participants requested this clarification
be included for the record... PGN]

You misinterpreted my message in a small way; I was writing about a university attended by a friend, NOT Rice university. To my knowledge, Rice has been very good about protecting its students' privacy. My student number is NOT my social security number, though the university has that number for good reasons. I do not want anyone to think that I was talking about Rice. David

---

Towards an effective definition of "autonomous" weapons

<LIN@XX.LCS.MIT.EDU>
Tue, 23 Sep 1986 18:00 EDT

[THE FOLLOWING DISCOURSE INVOLVING CLIFF AND HERB IS LIKELY
TO CONTINUE FOR A WHILE ON ARMS-D. PLEASE RESPOND TO HERB LIN,
NOT TO RISKS ON THIS ONE. HERB HAS VOLUNTEERED TO SUBMODERATE,
AND THEN SUBMIT THE RESULTS TO RISKS. PGN]

From: Clifford Johnson <GA.CJJ at Forsythe.Stanford.Edu>

An "autonomous weapon" [should be] defined to be any weapons system which is de facto preprogrammed to take decisions which, under the law of nations, require the exercise of political or military discretion.

It's not a bad first attempt, and I think it is necessary to get a handle on this. With the realization that you have done us a service in proposing your definition, let me comment on it.

I don't understand what it means for a weapon to "take a decision". Clearly you don't intend to include a depth charge set to explode at a certain depth, and yet a depth charge could "decide" to explode at 100 feet given certain input.

What I think you object to is the "preprogrammed" nature of a weapon, in which a chip is giving arming, targeting and firing orders rather than a human being. What should be the role of the human being in war? I would think the most basic function is to decide what targets should be attacked. Thus, one modification to your definition is

An "autonomous weapon" [should be] defined to be any weapons system which is preprogrammed to SELECT targets.

This would include things like roving robot anti-tank jeeps, and exclude the operation of LOW for the strategic forces.

But this definition would also exclude "fire-and-forget" weapons, and I'm not sure I want to do that. I want human DESIGNATION of a target but I don't want the human being to remain exposed to enemy fire after he has done so. Thus, a second modification is

An "autonomous weapon" [should be] defined to be any weapons system which is preprogrammed to SELECT targets in the absence of direct and immediate human intervention.

But then I note what a recent contributor said -- MINES are autonomous weapons, and I don't want to get rid of mines either, since I regard mines as a defensive weapon par excellence. Do I add mobility to the definition? I don't know.

Towards an effective defintion of "autonomous" weapons

Clifford Johnson <GA.CJJ at Forsythe.Stanford.Edu>
Monday, 22 September 1986 21:43-EDT

There's great difficulty in defining "autonomous weapons" so as to separate some element that seems intuitively "horrible" about robot-decided death. But a workable definition is necessary if, as CPSR tentatively proposes, such weapons are to be declared illegal under international law, as have chemical and nuclear weapons. (Yes, the U.N. has declared even the possession of nukes illegal, but it's not a binding provision.)

The problem is, of course, that many presently "acceptable" weapons already indiscrminately-discriminate targets, e.g. target-seeking munitions and even passive mines. Weapons kill, and civilians get killed too, that's war. Is there an element exclusive to computerized weapons that is meaningful?

I don't have an answer, but feel the answer must be yes. I proffer two difficult lines of reasoning, derived from the philosophy of automatic decisionmaking rather than extant weapon systems. First, weapon control systems that may automatically target-select among options based upon a utility function (point score) that weighs killing people against destroying hardware would seem especially unconscionable. Second, but this presumes a meaningful definition of "escalation," any weapons system that has the capability to automatically escalate a conflict - and is conditionally programmed to do so - would also seem unconscionable.

Into the first bracket would conceivably fall battle management software and war games, into the second would fall war tools that in operation (de facto) would take decisions which according to military regulations would otherwise have required the exercise of discretion by a military commander or politician. The latter category would embrace booby-trap devices activated in peacetime, such as mines and LOWCs; and here there is the precedent of law which prohibits booby traps which threaten innocents in peacetime. Perhaps the following "definition" could stand alone as *the* definition of autonomous weapons to be banned:

An "autonomous weapon" is defined to be any weapons system which is de facto preprogrammed to take decisions which, under the law of nations, require the exercise of political or military discretion.

This might seem to beg the question, but it could be effective - military manuals and international custom is often explicit on each commanders' degree of authority/responsibility, and resolving whether a particular weapon was autonomous would then be a CASE-BY-CASE DETERMINATION. Note that this could, and would, vary with the sphere of application of the weapons system. This is reasonable, just as there are circumstances in which blockades or mining is "legal" and "illegal."

Of course, a case-in-point would be needed to launch the definition. Obviously, I would propose that LOWCs were illegal. How about battle management software which decides to engage seemingly threatening entities regardless of flag, in air or by sea? Any other suggestions? Does anyone have any better ideas for a definition?

Towards an effective definition of "autonomous" weapons

<LIN@XX.LCS.MIT.EDU>
Tue, 23 Sep 1986 18:09 EDT

In thinking about this question, I believe that ARMS-D and RISKS could perform a real service to the defense community. There is obviously a concern among some ARMS-D and RISKS readers that autonomous weapons

are dangerous generically, and maybe they should be subject to some
legal restrictions. Others are perhaps less opposed to the idea.

It is my own feeling that autonomous weapons could pose the same danger to
humanity that chemical or biological warfare pose, though they may be
militarily effective under certain circumstances.

I propose that the readership take up the questions posed by Cliff's recent
contribution:

What is a good definition of an autonomous weapon?

What restrictions should be placed on autonomous weapons, and why?

How might such limits be verified?

Under what circumstances would autonomous weapons be militarily
useful?

Should we be pursuing such weapons at all?

How close to production and deployment of such weapons are we?

Maybe a paper could be generated for publication?

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 65

Wednesday, 25 September 1986

Contents

- UNIX and network security again
  Andy Freeman
- F-16 software
  Wayne Throop
- NYT feature article on SDI software
  Hal Perkins
- Autonomous widgets
  Mike McLaughlin
- Robottle Management Software?
  PGN
- Info on RISKS (comp.risks)

UNIX and network security again

Andy Freeman <ANDY@SUSHI.STANFORD.EDU>
Mon 22 Sep 86 17:09:27-PDT

preece%mycroft@gswd-vms.ARPA (Scott E. Preece) writes:

> If you can't trust the code running on another machine on your ethernet, then you can't believe that it is the machine it says it is, which violates the most basic principles of the NCSC model.

That's why electronic signatures are a good thing.

> I wrote (andy@sushi):
> \> Then NCSC certification means nothing in many (most?) situations.
>
> Well, most sites are required to have certified systems (yet?). If they were, they wouldn't be allowed to have non-complying systems.

The designers of the Ford Pinto were told by the US DOT to use $x as a cost-benefit tradeoff point for rear end collisions. Ford was still liable. I'd be surprised if NCSC certification protected a company from liability. (In other words, being right can be more important than complying.)

[This case was cited again by Peter Browne (from old Ralph Nader materials?), at a Conference on Risk Analysis at NBS 15 September 1986: Ford estimated that the Pinto gas tank would take $11 each to fix in 400,000 cars, totalling $4.4M. They estimated 6 people might be killed as a result, at $400,000 each (the going rate for lawsuits at the time?), totalling $2.4M. PGN]

Absolutely, network access should be as secure as phone access, IF YOU CHOOSE TO WORK IN THAT MODE. Our links to the outside world are as tightly restricted as our dialins. The Berkeley networking software is set up to support a much more integrated kind of network, where the network is treated as a single system. For our development environment that is much more effective. You should never allow that kind of access to a machine you don't control. Never. My interpretation of the original note was that the author's net contained machines with trusted-host access which should not have had such access; I contend that that represents NOT a failing of the software, but a failing of the administration of the network.

My interpretation of Brian's original message is that he didn't have a choice; Berkeley network software trusts hosts on the local net. If that's true, then the administrators didn't have a chance to fail; the software's designers had done it for them. (I repeated all of Scott's paragraph because I agree with most of what he had to say.)

-andy

[I think the implications are clear. The network software is weak. Administrators are often unaware of the risks. Not all hosts are trustworthy. The world is full of exciting challenges for attackers. All sorts of unrealistic simplifying assumptions are generally made. Passwords are typically stored or transmitted in the clear and easily readable or obtained -- or else commonly known. Encryption is still vulnerable if the keys can be compromised (flawed key distribution, unprotected or subject to bribable couriers) or if the algorithm is weak. There are lots of equally devastating additional vulnerabilities waiting to be exercised, particularly in vanilla UNIX systems and networks thereof. Remember all of our previous discussions about not trying to put the blame in ONE PLACE. PGN]

F-16 software

<rti-sel!dg_rtp!throopw%mcnc.csnet@CSNET-RELAY.ARPA>
Tue, 23 Sep 86 19:12:33 edt

> I spoke to an F-16 flight instructor about this business concerning bomb
> release when the plane is upside down. He said the software OUGHT to
> prevent such an occurrence. When the plane is not at the right angle of
> attack into the air stream, toss-bombing can result in the bomb being

> thrown back into the airplane.

Hmpf. *I* spoke to an ex Air-Force pilot. He said if *any* restriction on bomb release is incorporated it should be to prevent it when the plane (or more specificially, the bomb itself... there *is* a difference, and you had better realize it!) is pulling negative G's. This was my original point... "upside down" or "inverted" isn't the correct thing to worry about, it is the wrong mindset entirely, too simple a notion.

He went on to back up this assertion by pointing out that there is a common (well... well-known anyhow) bombing technique, called "over the shoulder" bombing, that requires release while inverted. Consider the following diagram. (Note that the trajectory shapes are unrealistic and the scales are exagerated. Limitations of the terminal, don't y'know.)

```
                     _
                    / \
                   /   \
          __________________________ |
         <         /     \r
                  /       \
                  |        |
                  v        /
B >_____________________________________/
                  T
```

Now, we have bomber B, release of bomb r, and target T. The bomber makes a fast, low-level run over the target (to avoid radar, and to let the bombsight get a good look). Then, soon after the overfly, pulls sharply up and over, and *while* *inverted* releases the bomb. The bomb lofts high into the air over the target whilst the plane scoots for home (rolling out of the inversion, presumably but not necessarily), and the bomb eventually lands splat on the target.

Basically, if you want the flight computer to wet-nurse the pilot at all in this regard, it ought to have a sensor to detect strain on the bomb restraints, and refuse to release them if the bomb isn't currently "trying" to "fall" away from the aircraft. (Even this isn't foolproof, of course, but it comes close.) Tying this into the *attitude* of the *aircraft* *itself* is *WRONG* *WRONG* *WRONG*, and is, as I said before, an architypical computer risk, in that it is an overly simple and misleading model of the situation.

The conversation I had with my friend makes a lot of sense to me, and the above somewhat vague stuff about the angle of attack does not. It could be I'm just missing something obvious, but I stand by my earlier position.

The desire for safety stands against every great and noble enterprise.
--- Tacitus

NYT feature article on SDI software

Hal Perkins <hal@gvax.cs.cornell.edu>

Wed, 24 Sep 86 11:32:59 EDT

The science section of last Tuesday's New York Times (16 Sept 1986) had a feature article on the SDI software problem starting on page C1. The headline is

Software Seen As Obstacle In Developing 'Star Wars'
Serious problems have forced dramatic changes in planning.

by Philip M. Boffey

The article is much too long to type in -- anyone interested can easily find a copy. The author has done his homework. He gives a good overview of the problems and of the issues in the SDI software debate and seems to have talked to the main people involved, several of whom are quoted. There's not much here that will be new to computer people who have been following the debate, but it's definitely worth reading.

Hal Perkins, Cornell CS

Autonomous widgets

Mike McLaughlin <mikemcl@nrl-csr>
Wed, 24 Sep 86 10:32:29 edt

The discussion of Autonomous Weapons should be expanded, considerably. Consider the following devices, soon to be found at your local dealer:

Autonomous Lumberjack - locates and cuts down designated trees (pulp, hardwood, diseased...)

Autonomous Booter - identifies automobiles with more than n dollars in overdue tickets.

Autonomous Streetsweeper - clears your street of any immobile object other than licensed vehicles (see A. Booter, above).

Autonomous NightWatchman - passive notifies authorities, active counteracts intruders.

N.B.: My "passive autonomous nightwatchman" is available at your friendly Heath/Zenith store _now_! Sorry, don't have a catalog at hand, or I'd provide ordering information.

Mike McLaughlin

[Mike, Now that it is FALL, you must be
feeling AUTUMNMATED. Autonomous Bosh]

Robottle Management Software? (Wine nought?)

Peter G. Neumann <Neumann@CSL.SRI.COM>

Wed 24 Sep 86 06:57:03-PDT

The following news item appeared in the 15 Sept 1986 issue of Digital Review, roundabout from the 26 June 1985 issue of the Halifax Gazette. But it is RISKY enough to report here.

> EDINBURGH (Reuters) -- A robot dressed in a black hat and bow tie appeared in court on Tuesday after running amok in a restaurant where it was employed to serve wine.
>
> Within its first hour on the job, the secondhand robot became uncontrollable, knocking over furniture, frightening customers and spilling a glass of wine, the court was told. The following day, the robot, exhibited Tuesday in the court, was still incapable of controlling the wine glasses, testimony said. Eventually its head fell into a customer's lap.

A tipsy-turvy robot? Did the firsthand know what the secondhand was doing? Asimov's Nth Law of Robotics might read, "A robot must not spill wine on the customers unless enforcing this Law would conflict with Laws 1,2, and 3." But maybe the program instructed the robot to put on "glasses" (ambiguously) so it could see better. Punishment: Send the robot to a OENAL COLONY? [Apologies in advance. I've been up too late recently.] Peter

---

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 66

Thursday, 25 September 1986

Contents

- Follow-up on Stanford breakins: PLEASE LISTEN THIS TIME!
  Brian Reid
- F-16 software [concluded?]
  Herb Lin
- Info on RISKS (comp.risks)

Follow-up on Stanford breakins: PLEASE LISTEN THIS TIME!

Brian Reid <reid@decwrl.DEC.COM>
25 Sep 1986 0014-PDT (Thursday)

"What experience and history teach is that people have never learned
anything from history, or acted upon principles deduced from it."
-- Georg Hegel, 1832

Since so many of you are throwing insults and sneers in my direction, I feel that I ought to respond. I am startled by how many of you did not understand my breakin message at all, and in your haste to condemn me for "asking for it" you completely misunderstood what I was telling you, and why.

I'm going to be a bit wordy here, but I can justify it on two counts. First, I claim that this topic is absolutely central to the core purpose of RISKS (I will support that statement in a bit). Second, I would like to take another crack at making you understand what the problem is. I can't remember the names, but all of you people from military bases and secure installations who coughed about how it was a network administration failure are completely missing the point. This is a "risks of technology" issue, pure and simple.

As an aside, I should say that I am not the system manager of any of the systems that was broken into, and that I do not control the actions of any of the users of any of the computers. Therefore under no possible explanation can this be "my fault". My role is that I helped to track the intruders down, and, more importantly, that I wrote about it.

I am guessing that most of you are college graduates. That means that you once were at a college. Allow me to remind you that people do not need badges to get into buildings. There are not guards at the door. There are a large number of public buildings to which doors are not even locked. There is not a fence around the campus, and there are not guard dogs patrolling the perimeter.

The university is an open, somewhat unregulated place whose purpose is the creation and exchange of ideas. Freedom is paramount. Not just academic freedom, but physical freedom. People must be able to walk where they need to walk, to see what they need to see, to touch what they need to touch. Obviously some parts of the university need to be protected from some people, so some of the doors will be locked. But the Stanford campus has 200 buildings on it, and I am free to walk into almost any of them any time that I want. More to the point, *you* are also free to walk into any of them.

Now let us suppose that I am walking by the Linguistics building and I notice that there is a teenager taking books out of the building and putting them in his car, and that after I watch for a short while, I conclude that he is not the owner of the books. I will have no trouble convincing any policeman that the teenager is committing a crime. More important, if this teenager has had anything resembling a normal upbringing in our culture, I will have no trouble convincing the teenager that he is committing a crime. Part of the training that we receive as citizens in our society is a training in what is acceptable public behavior and what is not. The books were not locked up, the doors to the library were not locked, but in general people do not run in and steal all of the books.

Or let me suppose instead that I am a reporter for the Daily News. I have a desk in a huge room full of desks. Most of the desks are empty because the other reporters are out on a story. You've seen scenes like this in the movies. It is rare in small towns to find those newsrooms locked. Here in Palo Alto I can walk out of my office, walk over to the offices of the Times Tribune a few blocks away, walk in to the newsroom, and sit down at any of those desks without being challenged or stopped. There is no guard at the door, and the door is not locked. There are 50,000 people in my city, and since I have lived here not one of them has walked into the newsroom and started destroying or stealing anything, even though it is not protected. Why not? Because the rules for correct behavior in our society, which are taught to every child, include the concept of private space, private property, and things that belong to other people. My 3-year-old daughter understands perfectly well that she is not to walk into neighbors' houses without ringing the doorbell first, though she doesn't quite understand why.

People's training in correct social behavior is incredibly strong, even among "criminals". Murderers are not likely to be litterbugs. Just because somebody has violated one taboo does not mean that he will immediately and systematically break all of them.

In some places, however, society breaks down and force must be used. In the Washington Square area of New York, for example, near NYU, you must lock everything or it will be stolen. At Guantanamo you must have guards or the Cubans will come take things. But in Palo Alto, and in Kansas and in Nebraska and Wisconsin and rural Delaware and in thousands of other places, you do not need to have guards and things do not get stolen.

I'm not sure what people on military bases use computer networks for, but here in the research world we use computer networks as the building blocks of electronic communities, as the hallways of the electronic workplace. Many of us spend our time building network communities, and many of us spend our time developing the technology that we and others will use to build network communities. We are exploring, building, studying, and teaching in an electronic world. And naturally each of us builds an electronic community that mirrors the ordinary community that we live in. Networks in the Pentagon are built by people who are accustomed to seeing soldiers with guns standing in the hallway. Networks at Stanford are built by people who don't get out of bed until 6 in the evening and who ride unicycles in the hallways.

Every now and then we get an intruder in our electronic world, and it surprises us because the intruder does not share our sense of societal responsibilities. Perhaps if Stanford were a military base we would simply shoot the intruder and be done with it, but that is not our way of doing things. We have two problems. One is immediate--how to stop him, and how to stop people like him. Another is very long-term: how to make him and his society understand that this is aberrant behavior.

The result of all of this is that we cannot, with 1986 technology, build computer networks that are as free and open as our buildings, and therefore we cannot build the kind of electronic community that we would like.

I promised you that I would justify what this all has to do with RISKS.

We are developing technologies, and other people are using those technologies. Sometimes other people misuse them. Misuse of technology is one of the primary risks of that technology to society. When you are engineering something that will be used by the public, it is not good enough for you to engineer it so that if it is used properly it will not hurt anybody. You must also engineer it so that if it is used *improperly* it will not hurt anybody. I want to avoid arguments of just where the technologist's responsibility ends and the consumer's responsibility begins, but I want to convince you, even if you don't believe in the consumer protection movement, that there is a nonzero technologist's responsibility.

Let us suppose, for example, that you discovered a new way to make screwdrivers, by making the handles out of plastic explosives, so that the screwdriver would work much better under some circumstances. In fact, these screwdrivers with the gelignite handles are so much better at putting in screws than any other screwdriver ever invented, that people buy them in droves. They have only one bug: if you ever forget that the handle is gelignite, and use the screwdriver to hit something with, it will explode and blow your hand off. You, the inventor of the screwdriver, moan each time you read a newspaper article about loss of limb, complaining that people shouldn't *do* that with your screwdrivers.

Now suppose that you have invented a great new way to make computer networks, and that it is significantly more convenient than any other way of making computer networks. In fact, these networks are so fast and so convenient that everybody is buying them. They have only one bug: if you ever use the network to connect to an untrusted computer, and then if you also forget to delete the permissions after you have done this, then people will break into your computer and delete all of your files. When people complain about this, you say "don't connect to untrusted computers" or "remember to delete the files" or "fire anyone who does that".

Dammit, it doesn't work that way. The world is full of people who care only about expediency, about getting their screws driven or their nets worked. In the heat of the moment, they are not going to remember the caveats. People never do. If the only computers were on military bases, you could forbid the practice and punish the offenders. But only about 0.1% of the computers are on military bases, so we need some solutions for the rest of us.

Consider this scenario (a true story). Some guy in the Petroleum Engineering department buys a computer, gets a BSD license for it, and hires a Computer Science major to do some systems programming for him. The CS major hasn't taken the networks course yet and doesn't know the risks of breakins. The petroleum engineer doesn't know a network from a rubber chicken, and in desperation tells the CS student that he can do whatever he wants as long as the plots are done by Friday afternoon. The CS student needs to do some homework, and it is much more convenient for him to do his homework on the petroleum computer, so he does his homework there. Then he needs to copy it to the CS department computer, so he puts a permission file in his account on the CSD computer that will let him copy his homework from the petroleum engineering computer to the CSD computer. Now the CS student graduates and gets a job as a systems programmer for the Robotics department, and his systems programmer's account has lots of permissions. He has long since forgotten about the permissions file that he set up to move his homework last March. Meanwhile, somebody breaks into the petroleum engineering computer, because its owner is more interested in petroleum than in computers and doesn't really care what the guest password is. The somebody follows the permission links and breaks into the robotics computer and deletes things.

Whose fault is this? Who is to blame? Who caused this breakin? Was it the network administrator, who "permitted" the creation of .rhosts files? Was it the person who, in a fit of expedience, created /usr/local/bin with 0776 protection? Was it the idiot at UCB who released 4.2BSD with /usr/spool/at having protection 0777? Was it the owner of the petroleum engineering computer? Was it the mother of the kid who did the breaking in, for failing to teach him to respect electronic private property? I'm not sure whose fault it is, but I know three things:

1) It isn't my fault (I wasn't there). It isn't the student's fault (he didn't know any better--what can you expect for $5.75/hour). It isn't the petroleum engineer's fault (NSF only gave him 65% of the grant money he asked for and he couldn't afford a full-time programmer). Maybe you could argue that it is the fault of the administrator of the CSD machine, but in fact there was no administrator of the CSD machine because he had quit to form a startup company. In fact, it is nobody's fault.

2) No solution involving authority, management, or administration will work in a network that crosses organization boundaries.

3) If people keep designing technologies that are both convenient and dangerous, and if they keep selling them to nonspecialists, then expedience will always win out over caution. Convenience always wins, except where it is specifically outlawed by authority. To me, this is one of the primary RISKs of any technology. What's special about computers is that the general public does not understand them well enough to evaluate the risks for itself.

F-16 software [concluded?]

<LIN@XX.LCS.MIT.EDU>
Thu, 25 Sep 1986 09:39 EDT

From: rti-sel!dg_rtp!throopw%mcnc.csnet at CSNET-RELAY.ARPA

> I spoke to an F-16 flight instructor about this business concerning
> bomb release when the plane is upside down. He said the software
> OUGHT to prevent such an occurrence. When the plane is not at the
> right angle of attack into the air stream, toss-bombing can result
> in the bomb being thrown back into the airplane.

Hmpf. *I* spoke to an ex Air-Force pilot. He said if *any* restriction on bomb release is incorporated it should be to prevent it when the plane (or more specificially, the bomb itself... there *is* a difference, and you had better realize it!) is pulling negative G's. This was my original point... "upside down" or "inverted" isn't the correct thing to worry about, it is the wrong mindset entirely, too simple a notion.

This dispute (well, sort of dispute anyway) is instructive -- each of us consulted our own experts, and we come away with different answers. It suggests why even defining safety is so hard. Maybe I misunderstood my flight instructor's response, or maybe I posed the question to him improperly, or maybe he just gave an off-the-cuff answer without thinking it thorugh, or maybe he's wrong...

Moral: When you are lost and ask for directions, never ask just one person for directions. Ask two people, and you have a better chance of getting to where you want to go.

Herb

[On the other hand, when the two people give you DIFFERENT DIRECTIONS, you must realize that AT LEAST ONE of them is wrong. So, you may have to ask THREE PEOPLE before you get any agreement... A further moral is that you should have some justifiable trust in those who are giving you advice. PGN]

Search RISKS using swish-e

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 67

Thursday 25 September 1986

Contents

- Old GAO Report on Medical Device Software
  Chuck Youman
- Re: Stanford breakin, RISKS-3.62 DIGEST
  Darrel VanBuer
- Re: Passwords and the Stanford break-in (RISKS-3.61)
  Dave Sherman
- Re: role of simulation - combat simulation for sale
  Jon Jacky
- MIT Symposium on economic impact of military spending
  Richard Cowan
- "Friendly" missiles and computer error -- more on the Exocet
  Rob MacLachlan
- Info on RISKS (comp.risks)

Old GAO Report on Medical Device Software

Chuck Youman <m14817@mitre.ARPA>
Thu, 25 Sep 86 13:49:08 -0500

I recently ran across an old report (Aug. 5, 1981) from the U.S. General Accounting Office (GAO) on the subject "Software Used in Medical Devices Needs Better Controls To Avoid Compromising Patient Safety (AFMD-81-95). I don't recall seeing it mentioned in this forum or in SEN. The report is 8 pages and can be ordered from the GAO at P.O. Box 6015, Gaithersburg, MD 20877.

To briefly summarize the report, they identified 78 cases involving unreliable computerized medical devices that occurred from June 1976 to August 1979. They state that the believe this is only a small fraction of the total cases that occurred. They examined 24 of the cases and found 13 of them had software problems. In their report they give two examples: a blood gas analyzer and a computerized electrocardiogram interpretation software package.

They concluded:

Advances in computer technology have brought about far more reliable hardware. However, software has been and remains a problem area, regardless of whether it is used in medical or business applications. We believe the use of software in medical devices is emerging as a troublesome area and requires the attention of the Bureau [i.e., the FDA].

The use of performance standards, as authorized by the Medical Device Amendments of 1976, is a possible mechanism to help control the performance of software in computerized medical devices. Unfortunately, the time-consuming process for developing standards together with the large number of standards to be developed makes it very unlikely that any standards will be available soon. This, coupled with the relatively fast pace at which computer technology changes, makes it unlikely that the standards when developed will be timely enough to validate software in medical devices. Therefore, we believe the Bureau needs to explore other alternatives for validating and certifying that the software in medical devices works as expected.

Charles Youman (youman@mitre.arpa)

Re: Stanford breakin, RISKS-3.62 DIGEST

Darrel VanBuer <hplabs!sdcrdcf!darrelj@ucbvax.Berkeley.EDU>
Wed, 24 Sep 86 09:35:37 pdt

I think many of the respondents misunderstand what went wrong: there was no failure in the 4.2 trusted networking code. It correctly communicated the message that "someone logged in as X at Y wants to run program Z at W". The failure of security was that

1) the "someone" was not in fact X because of some failure of security (e.g. poor password).
2) the real X who had legitimate access on W had previously created a file under some user id at W saying X at Y is an OK user.
3) the real X was lazy about withdrawing remote privileges (not essential, but widens the window of opportunity.

There's a tough tradeoff between user convenience in a networked environment and security. Having to enter a password for every remote command is too arduous for frequent use. Interlisp-D has an interesting approach:

1. Try a generic userid and password.
2. Try a host-specific userid and password.

In either case, if it does not have these items in its cache, it prompts the user. The cache is cleared on logout and at certain other times which suggest the user has gone away (e.g. 20 minutes without activity). Passwords are never stored in long term or publically accessible locations. It's also less convenient than 4.2 since you need to resupply IDs after every cache flush. It also has the opening for lazy users to use the same ID and password at every host so that the generic entry is enough.

Darrel J. Van Buer, PhD, System Development Corp., 2525 Colorado Ave

Santa Monica, CA 90406, (213)820-4111 x5449
...{allegra,burdvax,cbosgd,hplabs,ihnp4,orstcs,sdcsvax,ucla-cs,akgua}
!sdcrdcf!darrelj
VANBUER@USC-ECL.ARPA

Re: Passwords and the Stanford break-in (RISKS-3.61)

<mnetor!lsuc!dave@seismo.CSS.GOV>
Thu, 25 Sep 86 12:48:55 edt

There's another risk which isn't related to the problems of the networking code which Brian Reid described. Most users will have the same password on all machines. So where the intruder becomes root on one machine, he need merely modify login to store passwords for him, and will very quickly amass a collection of login-password combinations which have a very high probability of working all over the network.

I'm not sure what the solution is to this one, except, as has been pointed out, to be aware that the network is as vulnerable as its weakest link. Sure, people should use different passwords, but the burden of remembering passwords for many different machines can become onerous. Perhaps building a version of the machine name into the password can help mnemonically - i.e. use the same password with a different final letter indicating which machine it is.

I use two passwords for the several accounts I have: one for the machines under my control and one for guest accounts on other organizations' systems. That way no-one who collects passwords on someone else's system will be able to use them to break into Law Society machines.

Dave Sherman, The Law Society of Upper Canada, Toronto
dave@lsuc.UUCP
{ ihnp4!utzoo seismo!mnetor utai hcr decvax!utcsri } !lsuc!dave

[Mnemonics with one-letter differences are clearly easy to break. Also, it does not really matter how many passwords you have if they are stored somewhere for automatic remote access... The more realistic point is that network security is an intrinsically nontrivial problem. PGN]

Re: role of simulation - combat simulation for sale

Jon Jacky <jon@june.cs.washington.edu>
Thu, 25 Sep 86 17:10:09 PDT

I came across the following advertisement in AVIATION WEEK AND SPACE TECHNOLOGY, June 16, 1986, p. 87:

SURVIVE TOMORROW'S THREAT - <illegible> Equipment and Tactics Against Current and Future Threats

FSI's dynamic scenario software programs such as "War Over Land," "AirLand Battle," and "Helicopter Combat" provide realistic simulation of a combat environment. These programs use validated threat data to evaluate the effectiveness of individual weapons or an integrated weapons system. The easy-to-utilize programs are already in use by the Army, Navy, Air Force, and many prime defense contractors. Evaluate your system on a DoD-accepted model. For more information, contact ... (name, address, contact person).

(end of excerpt from ad)

The ad doesn't really say how you run this simulation, but kind of implies you can actually test real electronic warfare equipment with it. Needless to say, an interesting issue is, how comprehensive or realistic is this "validated (by whom? how?) threat data?" I checked the bingo card with some interest. And this ad is just one example of the genre - p. 92 of the same issue advertises a product called "SCRAMBLE! Full mission simulators," showing several high-resolution out-the-window flight simulator displays of aerial combat.

-Jonathan Jacky, University of Washington

MIT Symposium on economic impact of military spending

Richard A. Cowan <COWAN@XX.LCS.MIT.EDU>
Thu 25 Sep 86 17:42:50-EDT

[The following seminar, sponsored by MIT, may be of interest to RISKS Readers.]

November Symposium: "What are the effects of military spending?"
MIT Technology and Culture Seminar
Saturday, November 1, 1986
9am-3pm, MIT Room 26-100

Topics:

Bernard O'Keefe
--Chairman of the Executive Committee, EG&G, Inc.
"Are we focusing on the military confrontation with the USSR while ignoring the trade war with the Japanese?"

Seymour Melman,
--Professor of Industrial Engineering, Columbia University
"Do present rates of military spending make capital effectively available for civilian industry?"

Alice Tepper-Martin,
--Executive Director, Council on Economic Priorities
"If military spending is "only" about six or seven percent of the GNP, why worry?"

Frederick Salvucci
--Secretary of Transportation and Construction for Massachusetts
"Where will the funds for our national infrastructure come from?"

Barry Bluestone
 --Professor of Economics, Boston University
"The arms race and unemployment."

John Kenneth Galbraith
 --Professor of Economics, Harvard University
"Does the military-industrial complex really exist, and what is its impact?"

"Friendly" missiles and computer error -- more on the Exocet

Rob MacLachlan <RAM@C.CS.CMU.EDU>
Thu, 25 Sep 1986 21:23 EDT

[We have been around on this case in the past, with the "friendly" theory
having been officially denied. This is the current item in my summary list:
!!$ Sheffield sunk during Falklands war, 20 killed. Call to London
jammed antimissile defenses. Exocet on same frequency.
[AP 16 May 86](SEN 11 3)
However, there is enough new material in this message to go at it once
again! But, please reread RISKS-2.53 before responding to this. PGN]

I recently read a book about electronic warfare which had some things to say about the Falklands war incident of the sinking of the Sheffield by an Exocet missile. This has been attributed to a "computer error" on the part of a computer which "thought the missile was friendly." My conclusions are that:
1] Although a system involving a computer didn't do what what one might like it to do, I don't think that the failure can reasonably be called a "computer error".
2] If the system had functioned in an ideal fashion, it would probably have had no effect on the outcome.

The chronology is roughly as follows:

The Sheffield was one of several ships on picket duty, preventing anyone from sneaking up on the fleet. It had all transmitters (including radar) off because it was communicating with a satellite.

Two Argentinan planes were detected by another ship's radar. They first appeared a few miles out because they had previously been flying too low to be detected. The planes briefly activated their radars, then turned around and went home.

Two minutes later a lookout on the Sheffield saw the missile's flare approaching. Four seconds later, the missile hit. The ship eventually sank, since salvage efforts were hindered by uncontrollable fires.

What actually happened is that the planes popped up so that the could acquire targets on their radars, then launched Exocet missiles and left. (The Exocet is an example of a "Fire and Forget" weapon. Moral or not, they work.) The British didn't recognize that they had been attacked, since they believed that the Argentinans didn't know how to use their Exocet missiles.

It is irrelevent that the Sheffield had its radar off, since the missile skims just above the water, making it virtually undetectable by radar. For most of the flight, it proceeds by internal guidance, emitting no telltale radar signals. About 20 seconds before the end of the flight, it turns on a terminal homing radar which guides it directly to the target. The Sheffield was equipped with an ESM receiver, whose main purpose is to detect hostile radar transmissions.

The ESM receiver can be preset to sound an alarm when any of a small number of characteristic radar signals are received. Evidently the Exocet homing radar was not among these presets, since there would have been a warning 20 sec before impact. In any case, the ESM receiver didn't "think the missile was friendly", it just hadn't been told it was hostile. It should be noted that British ships which were actually present in the Falklands were equipped with a shipboard version of the Exocet.

If the failure was as deduced above, then the ESM receiver behaved exactly as designed. It is also hard to conceive of a design change which would have changed the outcome. The ESM receiver had no range information, and thus was incapable of concluding "anything coming toward me is hostile", even supposing the probably rather feeble computer in the ESM receiver were cable of such intelligence.

In any case, it is basically irrelevant that the ESM receiver didn't do what it might have done, since by 20 seconds before impact it was too late. The Sheffield had no "active kill" capability effective against a missile. Its anti-aircraft guns were incapable of shooting down a tiny target skimming the water at near the speed of sound.

It is also poossible to cause a missile to miss by jamming its radar, but the Sheffield's jamming equipment was old and oriented toward jamming russian radars, rather than smart western radars which wheren't even designed when the Sheffield was built. The Exocet has a large bag of tricks for defeating jammers, such as homing in on the jamming signal.

In fact, the only effective defense against the Exocet which was available was chaff: a rocket dispersed cloud of metalized plastic threads which confuses radars. To be effective, chaff must be dispersed as soon as possible, preferably before the attack starts. After the Sheffield, the British were familiar with the Argentinan attack tactics, and could launch chaff as soon as they detected the aircraft on their radars. This defense was mostly effective.

Ultimately the only significant mistake was the belief that the Argentinans wouldn't use Exocet missiles. If this possibility was seriously analysed, then the original attack might have been recognized. The British were wrong, and ended up learning the hard way. Surprise conclusion: mistakes can be deadly; mistakes in war are usually deadly.

I think that the most significant "risk" revealed by this event is tendency to attribute the failure of any system which includes a computer (such as the British Navy) to "computer error".

Search RISKS using swish-e

Report problems with the web pages to the maintainer

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 68

Friday, 26 September 1986

Contents

- VDU risks -- Government changes its mind, perhaps
  Stephen Page
- "Drive by wire" systems
  Charles R. Fry
- Viking Landers worked the first time and met the specs
  Dave Benson
- Unix breakins - secure networks
  David C. Stewart
- Comment on the reaction to Brian's Breakin Tale
  Dave Taylor
- Reliability, complexity, and confidence in SDI software
  Bob Estell
- Info on RISKS (comp.risks)

VDU risks -- Government changes its mind, perhaps

<Stephen Page <sdpage%sevax.prg.oxford.ac.uk@Cs.Ucl.AC.UK>>
Fri, 26 Sep 86 21:28:47 GMT

>From "Computer News" no. 141 (September 25, 1986):

Executive does U-turn on VDU risk

The [UK] government's Health and Safety Executive is spending nearly 1.5m pounds on research into the hazards of using VDUs -- just five months after assuring users that there is no danger.

The Executive has commissioned five reports into the possible health problems which may arise from working with VDUs.

The studies, which typically last three years, will look at topics such as repetitive VDU work, discomfort and optimum rest periods. It has contracted the work out to a number of universities at a cost of 475,000 pounds.

[...]

Earlier this year, the Executive issued a booklet aimed at dispelling fears that VDU work can lead to health risks and denying that radiation from terminals would lead to birth defects and miscarriages.

Part of the new research will look at the possible effects of VDU strain and stress on pregnant women.

[Of course, the US Government had previously cancelled some ongoing work in this area! PGN]

"Drive by wire" systems

Charles R. Fry <Chucko@GODZILLA.SCH.Symbolics.COM>
Tue, 23 Sep 86 08:59 PDT

From Henry Spencer:

Doug Wade notes:

> My comment to this, is what if a 8G limit had been programmed into
> the plane (if it had been fly-by-wire)...

My first reaction on this was that military aircraft, at least front-line combat types, obviously need a way to override such restrictions in crises, but civilian aircraft shouldn't. Then I remembered the case of the 727 ... It would seem that even [commecial] airliners might need overrides.

The "drive-by-wire" features now appearing in some cars, ostensibly to make them "safe to drive in all conditions," also seem to require overrides. For instance, the most common of these systems is anti-lock braking. The first such system available to the public, introduced by Audi on its original Quattro, could be disabled by a switch on the dashboard. Why? Because under some conditions (e.g. on gravel roads) the best braking performance is obtained when the wheels are locked. This was especially important on the Quattro, a street-legal rally car which was intended for high speed driving on all types of roads. (But as Detroit catches on, look for such switches to disappear in order to design some cost out of the systems.)

Now several European manufacturers (Mercedes-Benz, BMW) are introducing cars with "accelerative anti-skid systems," with no direct linkage between the gas pedal and the throttle on the engine. The intent is to prevent the engine from seeing full throttle when it would just cause excessive wheelspin, especially in slick, wintry conditions. However, on rear wheel drive cars (only!! -- don't try this with your Honda) such wheelspin can be used to make the car turn more tightly than it would without, and I can easily imagine circumstances in which this maneuver could save some lives.

No matter how many automated controls we install on cars (and airplanes) to prevent operators from exceeding their vehicles' limits, there will always be a need to allow the deliberate violation of these limits.

[Chuck added an aside on the value of high performance driving schools.]

-- Chuck Fry
Chucko@STONY-BROOK.SCRC.Symbolics.COM

Viking Landers worked the first time and met the specs

Dave Benson <benson%wsu.csnet@CSNET-RELAY.ARPA>
Wed, 24 Sep 86 18:01:18 pdt

Both Viking Landers worked in their first (and only) operation. The pre-operation testing simply ups one's confidence that the actual operation will be successful. Since the Viking Landers were the first man-made objects to land on Mars, Murphy's Law should suggest to any engineer that perhaps something might have been overlooked. In actual operation, nothing was.

Both Viking Mars shots had specifications for the length of time they were to remain in operation. While I do not recall the time span, both exceeded the specification by years. I do recall that JPL had to scrounge additional funds to keep the data coming in from all the deep-space probes, including the Vikings, as the deep space mechanisms were all working for far longer than expected.

Surely any engineered artifact which lasts for longer than its design specification must be considered a success. Nothing lasts forever, especially that most fragile of all artifacts, software. Thus the fact that the Viking 1 Lander software was scrambled beyond recovery some 8 years after the Mars landing only reminds one that the software is one of the components of an artifact likely to fail. So I see nothing remarkable about this event, nor does it in any way detract from judging both Viking Mars missions as unqualified engineering successes.

Unix breakins - secure networks

"David C. Stewart" <davest%tektronix.csnet@CSNET-RELAY.ARPA>
24 Sep 86 13:46:39 PDT (Wed)

One of the observations that have been made in the wake of the Stanford breakin is that Berkeley Unix encourages the assumption that the network itself is secure when in fact, it is not difficult to imagine someone tapping the ethernet cable and masquerading as a trusted host.

I have been intrigued by work that has been going on at CMU to support the ITC Distributed File System. (In the following, Virtue is the portion of the filesystem running on a workstation and Vice is that part running on the file server.)

The authentication and secure transmission functions are provided as part of a connection-based communication package, based on the remote procedure call paradigm. At connection establishment time, Vice and Virture are viewed as mutually suspicious parties sharing a common encryption key. This key is used in an authentication handshake, at the end of which each party is assured of the identity of the other. The final phase of the handshake generates a session key which is used for encrypting all further communication on the connection. The use of per-session encryption keys reduces the risk of exposure of authentication keys. [1]

The paper goes on to state that the authorization key may be supplied by a password (that generates the key but is not sent along the wire in cleartext) or may be on a user-supplied magnetic card.

This is one of the few systems I have seen that does not trust network peers implicitly. A nice possibility when trying to reduce the risks involved with network security.

Dave Stewart - Tektronix Unix Support - davest@tektronix.TEK.COM

[1] "The ITC Distributed File System: Principles and Design", Operating Systems Review, 19, 5, p. 43.

Comment on the reaction to Brian's Breakin Tale

Dave Taylor <taylor%hpldat@hplabs.HP.COM>
Fri, 26 Sep 86 17:55:53 PDT

I have to admit I am also rather shocked at the attitudes of most of the people responding to Brian Reids' tale of the breakin at Stanford. What these respondents are ignoring is The Human Element.

Any system, however secure and well designed, is still limited by the abilities, morals, ethics, and so on of the Humans that work with it. Even the best paper shredder, for example, or the best encryption algorithm, isn't much good if the person who uses it doesn't care about security (so they shred half the document and get bored, or use their husbands' first name as the encryption key).

The point here isn't to trivialize this, but to consider and indeed, PLAN FOR the human element.

I think we need to take a step back and think about it in this forum...

-- Dave

Reliability, complexity, and confidence in SDI software

"ESTELL ROBERT G" <estell@nwc-143b.ARPA>

26 Sep 86 13:22:00 PST

I apologize in advance for the length of this piece. But it's briefer than the growing list of claims and counter-claims, made by resepctable folks, based on either/both sound theory or/and actual experience. And we're dealing with a critical question:
Can very large systems be reliable?

The "bathtub curve" for MECHANICAL "failures" has always made sense to me. I've heard lectures about how software follows similar curves. But I've really been stumped by the notion that "software wears out."

I'd like to attempt to "bound the problem" so to speak. SUPPOSE that we had a system composed of ten modules; and suppose that each module had ten possible INTERNAL logical paths, albeit only one entry and only one exit.

The MINIMUM number of logical paths through the system is ten (10); i.e., *IF* path #1 in module A INVARIABLY invokes path #1 in modules B, C, ... J; and likewise, path #2 in A INVARIABLY invokes path #2 in B, C, ... J; etc. then there are only ten paths.
NOTE I'm also assuming that the modules invariably run in alpahbetical order, always start with A, and always finish with J; and never fail or otherwise get interrupted. [I'm trying to avoid nits.]
Some residential wiring systems are so built; there are many switches and outlets on each circuit; but each circuit is an isolated loop to the main "fuze" box; "fuzes" for the kitchen are independent of the den.

The MAXIMUM number of logical paths through the system is ten billion (10.E10); i.e., *IF* each module can take any one of its ten paths in response to any one of the ten paths from any one of the other ten modules, there are 10**10 possibilities.
AGAIN assuming that the system always starts with A, runs in order, etc.
*IF SEQUENCE IS SIGNIFICANT, and if the starting point is random, THEN there are ten!10.E10 paths; i.e., ten factorial times ten billion, or 36,288,000,000,000,000 possible paths in the system.

Further, *IF INTERRUPTS* are allowed and are significant, then I can't compute the exact number of possible paths; but I can guarantee that it's >MORE> than 10!10.E10.

End of bounds. The scope reaches from the trivial, to the impossible.

The GOAL of good engineering practices [for hardware, software, and firmware] is to design and implement modules that control the possible paths; e.g., systems should *NOT* interact in every conceivable way.
It does NOT follow that the interactions should be so restricted that there are only ten paths through a ten module system.
BUT there is some reason to HOPE that systems may be so designed, in a tree structure such that:

a. AT EACH LEVEL, exactly one module will be "in control" at any instant;
b. and that each module will run independently of others at its level;

c. and that there are a finite [and reasonably small] number of levels.

In "Levels of Abstraction in Operating Systems", RIACS TR 84.5, Brown, Denning, and Tichy describe 15 levels, reaching from circuits to shell; applications sit at level 16. If one must have a layered application, then add layers 17, 18, et al.

I will conjecture that at levels 1 and 2 [registers, and instruction set], there are only five possible states (each):
(1) not running;
(2) running - cannot be interrupted;
(3) running - but at a possible interrupt point;
(4) interrupted; and
(5) error.

I will further conjecture that the GOAL of writing modules at each of the other layers, from O/S kernel, through user application packages, can reasonably be to limit any one module to ten possible states. NOTE that purely "in line code" can perform numerous functions, without putting the module in more than a few states. [e.g., Running, Ready to run, Blocked, Intrerrupted, Critical region, or Error.]

Such a system, comprised of say 15 applications layers, would assume maybe 290 possible states; that's the SUM of the number of possibilities at each layer, given the path that WAS ACTUALLY TAKEN to reach each layer.

Yet the number of functions that such a system could perform is at least the sum of all the functions of all the modules in it. If you're willing to risk some interaction, then you can start playing with PRODUCTS [vice SUMS] of calling modules, called modules, etc. EVEN SO, if the calling module at layer "n" can assume half a dozen states, and the called module at layer "n+1" can assume a similar number, then the possible states of that pair are about 40; that's more than a dozen, but it's still managable.

In real life, both humans and computers deal with enormously complex systems using similar schemes. For instance, two popular parlor games: chess, and contract bridge. Each admits millions of possible scenarios. But in each, the number of possible sensible *NEXT plays* is confined by the present state of affairs. So-called "look ahead" strategies grow very complex; but once a legal play has been made, there are again a small number of possible legal "next plays."

In bridge, for instance, at least 635,013,559,600 possible hands can be dealt, to ONE player [combination of 52 things, 13 at a time]. That one hand does not uniquely determine the contents of the other three hands. Whether the hands interact is not a simple question in pure mathematics; in many cases, they do; but in one unique case, they don't; e.g., if dealer gets all 4 aces, and all 4 kings, all 4 queens, and any jack, then he bids 7 no trump; and it doesn't matter who else has what else; it's an unbeatable bid. [Non bridge players, accept both my word for it; and my apology for an obscure example.]

We've been playing bridge a lot longer than we've been writing large, real-time software systems. I'll conjecture that we don't know nearly as much

about "SDI class systems" as we do about the card game.
But in either case, if we aren't careful, the sheer magnitude of the numbers can overwhelm us.

BOTTOM LINEs:

1. The curve for debugging software has a DOWNslope and length that is some function of the number of possible paths through the code.

2. Good software engineering practice says that one checks the design before writing lots of code. ["Some" may be necessary, but not "lots."]
IF errors show up in the design, fix them there.
IF the DESIGN itself is flawed, then change it. [e.g., Rethink a design that allows modules to interact geometrically.]

3. Confidence builds as one approaches the 90% [or other arbitrary level] point in testing the number of possible paths.

4. The reason that we haven't built confidence in the past is that we've often run thousands of hours, without knowing either:

a. how many paths got tested; or
b. how many paths remained untested.

5. INTERACTIONS do occur - even ones that aren't supposed to.
[Trivial example: My car's cooling and electrical systems are NOT supposed to interact; and they don't - until the heater hose springs a leak, and squirts coolant all over the distributor and sparkplugs.]
In "The Arbitration Problem", RIACS TR 85.12, Dennning shows that computers are fundamentally NOT ABSOLUTELY predictable; it may be that an unstable state is triggered ONLY by timing idiosyncracies such as:
At the same minor cycle of the clock, CPU #1 suffers a floating underflow in the midst of a vector multiplication, AND CPU #2 takes an I/O interrupt from a disk read error, while servicing a page fault.

6. Since interactions do occur, experiences that many have had with small programs in a well-confined environment do *NOT* necessarily "scale up" to apply to very large, real-time codes, that run on raw hardware in a hostile [or just "random"] environment. NOTE that I'm claiming that in such a system, the O/S kernel is part of the real-time system.

7. The "problem space" we've been discussing is at least triangular.
In one corner, there are assembly language monoliths, running on second-generation computers, without hardware protection; such systems convince Parnas that "SDI won't ever work." Written that way, it won't.
[Important aside: It's one thing to argue that *if* SDI were built using modern software techniques, it would work. It's another thing to realize that in DOD, some (not all) tactical systems run on ancient computers that cost more to maintain than they would to replace; and offer less power than a PC AT. Such facts, known to Parnas, understandably color his thinking.]

In another corner, there are small [1000 or so lines] modules, running in a controlled environment, that and have been "proven" to work.

Most of us doubt that such experience scales up to SDI sizes.

In another corner, there are 100,000 line systems that work, in real life,
but without formal proofs. Probably built using good S/W Eng practices.

8. The KISS principle ["Keep It Simple, Stupid"] eliminates lots of problems.
Prof. Richard Sites, at UCSD in 1978, told of a talk given by Seymour Cray.
In answer to audience questions about "how to make the circuits run at those speeds", Cray explained that circuit paths were all of known, fixed lengths; and that all paths were terminated cleanly at both ends; and other "good EE practices" taught to undergrads. Less successful builders were so wrapped up in megaFLOPS that they got careless.

We could do well to adopt Cray's philosophy for hardware as we build our software; e.g., build "RISC" programs; write code that does only a few tasks, but does them very well, both quickly and reliably.
Maybe that's one reason why UNIX systems are so portable, powerful, and popular? [Each module is simple; power comes from piping them together.]
NOTE that I'm claiming that "RISC" computer architecture is not new; look at almost every machine that Cray has designed; instruction sets are limited, and their implementation is superb.

Bob
For the record, I'm speaking "off the record" and expressing personal opinion.

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 69

Sunday, 28 September 1986

Contents

- Confidence in software via fault expectations
  Dave Benson
- More on Stanford's UNIX breakins
  John Shore
  Scott Preece
- F-16 simulator
  Stev Knowles
- Deliberate overrides?
  Herb Lin
- Viking Landers -- correction to RISKS-3.68
  Courtenay Footman
- Info on RISKS (comp.risks)

Confidence in software via fault expectations

Dave Benson <benson%wsu.csnet@CSNET-RELAY.ARPA>
Sat, 27 Sep 86 18:14:48 pdt

A partial reply to Estell's nice piece on "Reliability, complexity
and confidence in SDI software" as well as other comments about
fault rates in large software:

(1) The bathtub curve for reliability of engineered artifacts is usually
considered to be composed of three distinct phenomena,
   (i) The early failures caused by manufacturing defects,
   (ii) The "random" failures of components for "unknown" reasons
      (These may be judged as defects in the design, allowed
       to lower the cost of the product),
   (iii) Wearout failures near the end of the product life.
Type (i) failures give the initial declining failure rates during "burn-in",
type (ii) failures during the useful product life, and type (iii) failures
occur at the design-life limit. This bathtub curve is not applicable to
software since the usual definition of a large software product includes
many different releases. Perhaps a software product should be compared to

several different models of the same car, toaster, airplane, etc. The bathtub curve describes the sum of manufacturing defects, design defects, and wear. Software ordinarily has no manufacturing defects and the usual way ordinary backups are done insures that most software does not wear out before it becomes obsolete. Perhaps the Viking 1 Lander software failure could be classified as a "wearout" due to inadequate preventative maintenance, but this seems to be streching a point.

So software ordinarily fails from design defects and design defects only. These are considered so important that we classify such defects into specification, design and implementation defects. The point here is that none of these are manufacturing or wear failures.

(2) The defect rate models for software all attempt to describe a process of redesigning the software after the discovery of failures, repeatedly, in a never-ending cycle of testing (either formally or via users discovering problems) and "maintenance" (which is actually redesigning a new model of the software upon discovery of problems--with so-called enhancements thrown in to confuse the issues). I shall now give a crude approximation to all of these models. Let all realize I have abstracted the essential features of these models to the point of unusability in QA practice. The essense is enough to make my point.

We assume that the original release of the software has a load of N design defects and that defects are discovered and instantly and flawlessly reworked with a rate constant, a, according to the formula

R(t) = N*exp(-a*t)

where exp() is the exponential function, t is a measure of software use (time, person-years, cpu cycles consumed, ...) and R(t) is the remaining number of design faults in the reworked software. This formula clearly illustrates that for any t>=0, if R(t) is not zero, then more faults remain. In words, some faults mean yet more faults.

The more detailed versions of this essential idea do, approximately, describe the process of removing faults from a continuing sequence of releases of a software product. Bev Littlewood has a nice survey of these, together with some practical suggestions, in a recent IEEE Trans. on Software Engineering--perhaps last Jan or Feb issue. In any case, we may see that the essential feature of "some faults imply more faults" is used in practice to estimate remaining design fault loads in software. The models have this feature because this seems theoretically sound and the actual data is not inconsistent with this class of models.

(3) If faults are not repaired when discovered, there is data suggesting that software failures may be viewed as type (ii), supra: Singpurwalla and Crow have a nice paper suggesting that faults are evidenced as failures with a periodicity sufficiently good to make interesting Fourier analysis of the failure data. We may take this as suggesting that some failures imply more failures at regular times in the future.

(4) Good designs have few faults and evidence few failures. In software this means few releases are necessary to correct faults. However, many software products interact primarily with that most flexible of io devices, people, People quickly adjust to the ideosyncracies and failures of the software they use. In my opinion, Unix (Reg. Trademark, AT&T) and derivatives is successful because its ideosyncracies and failures are somehow "human", but not because of low failure rates.

Good software designs start with a low initial number of faults. Good design practices seem to lead to better software. But one simply requires more data than currently exists to say much definite about the advantages of Ada vs. a more traditional practice. Furthermore, new software is likely to be "more complex" than old software--leading to perhaps the same MTTF. Highly reliable software appears to be engineered in much the same manner as any other highly reliable engineered artifact: By repeatedly designing similar artifacts, obtaining experience with their use, and then redesigning anew.

(5) Thus many of us are extremely dubious about the claims made for SDI (and thus its driving software). Without the ability to test in actual practice, there is no compelling reason to believe any claims made for the reliability of the software. This point has been made several times, by several people, on RISKS and I'll not repeat the argument. It seems that the onus of compelling evidence lies with those who claim SDI "will work." So far I've found no evidence whatsoever to support the claim that ANY new military software works in its first adversarial role: i.e., in the face of enemy action or a good, determined, simulation thereof. I'd appreciate reliable evidence for such. The claim for 100,000 line programs which work reliably requires supporting evidence. I am perfectly prepared to believe that the 28th yadbm (yet another data base manager) works reliably. I'm not prepared to simply accept such claims for military software. An example: JSS is a C3I system for the defense of North America against bomber attack. JSS is currently receiving some kind of "independent operational" test in Colorado. Workers at Hughes kept careful records of defect rates during development, and reported that certain of the standard models alluded to above failed to predict defect rates at the next step of in-house testing. Will I ever be able to learn what the results of the "independent operational" test are? I doubt it. All I might be able to learn is whether the US adopts the system or not. I'm highly dubious about the reliability of JSS, despite the adoption of reasonably current SE practices. And recall, JSS is the nth yac3i.

(6) Controlling complexity is a wonderful idea. But what does one do in the face of a truely complex world, in which complex decisions must be made? One designs complex software. Recall that the Enroute Air Traffic Control System has so far exercised only a minute fraction of all the paths through it, despite being installed at about 10 sites for about 10 years. At the current rate one might get to 90% path coverage by the year 2200? Yet every time you fly on a commercial aircraft, you implicitly trust this system. I suggest you trust it because it has been used operationally for 10 years and the enroute controllers view it as trustworthy. The fault rate is low enough and the controllers flexible enough and the enroute mid-air near collision rate is low enough that everyone is satisfied enough. No mathematics and little statistics here--just actual operational experience.

(7) Software types need to adopt rather more of a Missouri attitude: Show me that it works. Part of the problem is defining what "works" means. Thats what makes the Viking Lander experiences so compelling.

Everyone can easily agree that the software worked the only two times it was called upon to land the craft. One might think that military software experiences should be equally compelling to the senses. So consider the Navy's Aegis experiences... The result of actual data suggests that SDI software is unbuildable as a highly reliable program. I repeat my call for serious, professional papers on military software which worked the first time. So far I can only conclude than none such exist. Thereby I think I am entitled to discount any claims for the "quality" of military software in actual, operational practice. The logical, rational conclusion is that, with no data supporting claims for military software working in first use, and only data such as the Sgt. York and Aegis, SDI software will not work the first and only time it might be called upon to function.

Re: Brian Reid's follow-up on Stanford's UNIX breakins

John Shore <epiwrl!shore@seismo.CSS.GOV>
28 Sep 86 10:51:16 EST (Sun)

Brian is quite right. The job of an engineer is to build systems that people can trust. By this criterion, there exist few software engineers.

js

Follow-up on Stanford breakins: PLEA

"Scott E. Preece" <preece%ccvaxa@GSWD-VMS.ARPA>
Fri, 26 Sep 86 10:26:33 cdt

Brian Reid speaks eloquently to important issues. Virtually everything he says in this note makes perfect sense and should be taken to heart by everyone designing systems. BUT...

What he says now is not exactly what he said the first time; when, I assure him, some of us were listening. His first note did in fact attribute blame, to the networking code and to the student involved (under the general rubric of 'wizards').

The designer of the gelignite-handled screwdriver has clearly got a responsibility when the screwdriver is (incorrectly) used to pound on something and explodes. The designer has little responsibility when the screwdriver is used (incorrectly and maliciously) as a sharp object to stab a co-worker during a fight. If the screwdriver is used to hit someone over the head in a fight, and explodes, the responsibility is a lot more muddled. It is not at all clear how far the designer's responsibility for protecting us from mistakes extends to protecting us from temptation.

Is a car manufacturer morally liable for its cars being capable of going 120 mph, creating the potential for more serious accidents when they are used inappropriately? Is the manufacturer of autodial modems responsible because they make it possible for system crackers to try many more phone numbers per hour than manually dialled modems?

Had Brian made slightly less attempt to de-jargonize his original posting and said ".rhosts" instead of "permission files", which could refer to quite a few different things ina BSD system, I would have taken a different impression of his complaint away from that original posting. I agree strongly that .rhosts files are a danger that administrators should be able to turn off, preferably on a host by host basis.

It should still be noted that .rhosts files are there for a reason and that that reason is perfectly valid and the provision of .rhosts capabilities perfectly reasonable IN THE APPROPRIATE SITUATION. A campus-wide network of machines under diverse administrators may not be such a situation; I would hate to see the capabilities taken out of the system simply because there may be inappropriate situations. Ftp and telnet are still provided as well as the r-utilities.

As our moderator has said, fault rarely lies on one head. I agree with Brian that the designer (of systems OR screwdrivers) has a strong responsibility to consider both unintentional and intentional misuses of her systems and to watch for aspects of her designs that could raise the consequences of such misuses. The strongest responsibility is to make the limits of appropriate use obvious to the user, by packaging, documentation, and whatever other steps may be necessary. If on mature reflection it still seems likely the user will be unaware of the problem (who reads documentation on a screwdriver), the designer has a moral obligation to seek other means to avoid misuse. Perhaps the explosive screwdriver should be sold only with with a two-foot long handle, making it unsuitable for common domestic use, or as a separately packaged replacement handle in a six-inch thick lead box bedecked with scenes of mutilation. If, however, the object is the best or only solution to a particular problem (only a gelignite screwdriver can remove red kryptonite screws from lead doorframes), it may also be morally unacceptable to suppress the product simply because it may have dangerous implications in the hands of the unwary.

Hey, surprise, there's no easy answer...

scott preece, gould/csd - urbana, uucp: ihnp4!uiucdcs!ccvaxa!preece

[Let me commend Brian once again for having performed a truly valuable service to the community. (I notice his original message is reappearing in many places!) I don't think we should expect him to try to respond to each such comment. But -- given the ease with which system and network security can be broken -- we may see lots more of such analyses of OTHER breakins. The sad part is that most of these vulnerabilities are well known in the security community, but few other people have yet been concerned enough to do anything, including most system developers. The consensus among security folks is that it will take a Chernobyl-like event in computer security before most people wake up. PGN]

F-16 simulator

Stev Knowles <stev@BU-CS.BU.EDU>
Thu, 25 Sep 86 17:36:43 EDT

As I see it, you are all missing the point. A simulator *should* allow the plane to land with the gear up. A simulator should allow it to release a bomb in any position, *if the plane would*. The simulator should not try and stop the pilot from doing stupid things, it should react as the plane would. *If the plane will not allow something*, then the simulation should not allow it.

There is a difference. the *plane* should not allow a bomb to be detached if it will damage the plane. *But if it does* the software should too.

stev knowles, boston university distributed systems group
CSNET: stev@bu-cs.CSNET UUCP:...harvard!bu-cs!stev BITNET:ccsk@bostonu.BITNET

Deliberate overrides?

<LIN@XX.LCS.MIT.EDU>
Sat, 27 Sep 1986 08:36 EDT

> From: Charles R. Fry <Chucko at GODZILLA.SCH.Symbolics.COM>
> No matter how many automated controls we install on cars (and airplanes) to prevent operators from exceeding their vehicles' limits, there will always be a need to allow the deliberate violation of these limits.

This discussion about allowing overrides to programmed safety limits worries me. It is certainly true that there are instances in which the preservation of life requires the operator to override these devices. But these have to be weighed against the situations in which a careless operator will go beyond those limits when it is inappropriate. I haven't heard much discussion about that, and maybe it is because it is very difficult (impossible?) for the safety machinery to tell when an operator is being careless given the operative conditions at the time.

There is a tradeoff here that many have resolved categorically in favor of people being able to override computers. I think only competent and sensible people people, under the right circumstances, should be able to do so. The problem is to find a mechanical system capable of making these distinctions. Thus, the comment that PGN omitted "[Chuck added an aside on the value of high performance driving schools.]" was, in my view, crucial to understanding the situation involved. Maybe a partial solution would be to allow only drivers who have passed courses at high performance driving schools to override.

Herb [Shades of Chernobyl! PGN]

Viking Landers -- correction to RISKS-3.68

Courtenay Footman <cpf@tcgould.tn.cornell.edu>

Sun, 28 Sep 86 22:12:58 EDT

>Date: Wed, 24 Sep 86 18:01:18 pdt
>From: Dave Benson <benson%wsu.csnet@CSNET-RELAY.ARPA>
> ... Since the Viking Landers were the
>first man-made objects to land on Mars, ...

Actually, the first man-made object to land on Mars was a Russian craft that sent about 30 seconds of carrier signal and then died. Nobody knows exactly what happened to it.

Courtenay Footman ARPA: cpf@lnsvax.tn.cornell.edu
Lab. of Nuclear Studies Usenet: cornell!lnsvax!cpf
Cornell University Bitnet: cpf%lnsvax.tn.cornell.edu@WISCVM.BITNET

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 70

Monday, 29 September 1986

Contents

- Deliberate overrides?
  Scott E. Preece
- Multiple causes and where to place the "blame"
  PGN
- The Art of "Science" and its Computers
  PGN
- No-lock Brakes
  Peter Ladkin
- Sanity in Automating Keyword Abstracting
  Brint Cooper
- The Network Is Getting Old?
  PGN
- Info on RISKS (comp.risks)

Deliberate overrides?

"Scott E. Preece" <preece%ccvaxa@GSWD-VMS.ARPA>
Mon, 29 Sep 86 09:17:27 cdt

/**** ccvaxa:fa.risks / LIN@XX.LCS.MI / 2:47 am Sep 29, 1986 ****/
> From: Charles R. Fry <Chucko at GODZILLA.SCH.Symbolics.COM>
>
> > No matter how many automated controls we install on cars (and airplanes)
> > to prevent operators from exceeding their vehicles' limits, there will
> > always be a need to allow the deliberate violation of these limits.

> From: LIN@XX.LCS.MIT.EDU

> This discussion about allowing overrides to programmed safety limits
> worries me.

One of the nice things about computer-driven controls, as opposed to mechanical controls, is that they allow you to be less draconian in specifying limits. You don't have to build a bomb release that can never,

ever allow the pilot to drop a bomb while inverted; you can instead say "You know, if I do what you've asked, the bomb is going to fall on the wing and probably strip off your starboard control surfaces." and the pilot can say "Yes, I know, do it anyway." And by providing a (safety-covered and hard-to-reach) button that says "Override control limits" you can even make it possible for the pilot to say in advance that at this point she feels the danger in overriding the controls is smaller than the danger in not overriding the controls.

The reason we think it's reasonable to require automated controls to allow exceptions is that we know the automated controls have allowed and encouraged us to incorporate limits and we recognize that (1) those limits may have erred on the side of normal safety, (2) since the systems are new, the necessary operational envelope may not be known, and (3) the interaction of the limits may create unanticipated problems. Yes, users should be allowed to override automated controls in almost all cases AND designers should make very, very sure that the effort to override is proportional to the danger of the override. In many cases there should also be logging of overrides, so that operators, maintainers, and designers have an opportunity to notice that actual use seems to be violating the design assumptions.

I wonder how many readers of this list [NO, this is NOT a survey, DON'T write to tell me] drive cars with manual transmissions precisely because they want to be in control, want to know that doing x and y will result in the car doing z, without any control system in the way to place limits on actions or responses...

scott preece gould/csd - urbana uucp: ihnp4!uiucdcs!ccvaxa!preece

---

Multiple causes and where to place the "blame"

Peter G. Neumann <Neumann@CSL.SRI.COM>
Mon 29 Sep 86 21:36:04-PDT

Today's AP noted that the FAA may cite the pilot of the Grumman Yankee for being in restricted airspace, at precisely the moment of the crash between the Aeromexico jet and the Piper Archer (which was also in restricted airspace), which distracted the air traffic controller from attending to the jet and the Piper (the absence of whose altitude information was also a factor) -- at precisely the time the crash occurred. The controller did tell the Grumman pilot that he was in restricted air space, but then granted him permission to continue (and that negotiation took two precious minutes away from his attention to the jet).

The Art of "Science" and its Computers

Peter G. Neumann <Neumann@CSL.SRI.COM>
Mon 29 Sep 86 16:36:51-PDT

A computer of the AAAS sent out renewal bills for SCIENCE to some subscribers:
    Subscription price $6647, Postage $732,

Voluntary contribution $10, Total $5437.
The subscription price during 1986 was $60, and the accompanying letter from the president of AAAS noted that inflation had required an increase. A quite amusing editorial by Daniel Koshland, Jr. in the 26 Sept 86 issue wondered whether any people would rush out to take advantage of the incorrect addition.

No-lock Brakes

Peter Ladkin <ladkin@kestrel.ARPA>
Mon, 29 Sep 86 14:47:16 pdt

A minor correction to Chuck Fry's comments - the first anti-skid system on a production car was installed on a Jensen (pre-dating the Jensen-Healey) in the 60s. It was made by Lockheed, and derived directly from aircraft systems.

Sanity in Automating Keyword Abstracting

Brint Cooper <abc@BRL.ARPA>
Mon, 29 Sep 86 15:09:02 EDT

Here is an example of a risk associated with the use of computers. The risk is to the accurate dissemination of information and is caused by faulty programming (programmers?).

Today, the BRL Librarian informed us that the Defense Technical Information Center (DTIC, formerly known as DDC) now requires that the titles of our technical reports (the principal products of a research lab such as the BRL) be written so that the "keywords" are found in the first five words of the title.

Thus, a report which formerly was titled "Communication Modeling in the Artillery Control Experiment" with keywords "error control," "tactical communications," "networks," and "modeling" would have to be titled "Modeling, Tactical Communications, and Error Control Networks," thus sounding like, as one chap here put it, "a four volume set by Harry van Trees" instead of a 25 page report.

Exact text of our librarian's notice follows:

> We have been advised by DTIC that the titles of technical reports should
> be designed with the key words positioned in the first five words of the
> title. This is because only the first five words are used in a title
> search in the DTIC electronic data base DROLS.(DEFENSE RESEARCH ON LINE
> SYSTEM). Important to know (and remember) is that articles are counted
> in those first five words. Therefore a report entitled "A report of the
> effect........" will not have any key words picked up in a title
> search. If you currently have a report in editing, we will review it
> and if it it does not comply with the DTIC recommendation we will advise
> you so that it can be reworked. If you currently have a report under
> review or in writing you might like to think about a title change.
> Please give this widest possible dissemination.

Brint
ARPA: abc@brl.arpa UUCP: ...{seismo,unc,,decvax,cbosgd}!brl-smoke!abc

Dr Brinton Cooper, U.S. Army Ballistic Research Laboratory
Attn: SLCBR-SE-C (Cooper), Aberdeen Proving Ground, MD 21005-5066
Offc: 301-278-6883 AV: 298-6883 FTS: 939-6883 Home: 301-879-8927

[ASK NOT WHAT YOUR COMPUTER CAN DO FOR YOU,
ASK WHAT YOU CAN DO FOR YOUR COMPUTER!
I started to add a diatribe, but gave up in annoyance. PGN]

The Network Is Getting Old?

Peter G. Neumann <Neumann@CSL.SRI.COM>
Mon 29 Sep 86 09:50:22-PDT

There has been an enormous amount of difficulty in dealing with the ARPANET since early September, perhaps related to the installation of new IMP software and subsequent patches when the new release did not work properly. I had devastating problems TELNETing from four different East-coast hosts to three different SRI systems (irrespective of whether there was a gateway at my end, and even with no loads on either system). No answers have been forthcoming from any of our gurus, so the problems remain pervasive and painful. I am also getting a rash of returned net-barfed RISKS mail (as well as RISKS filling up peoples' directories when they go on vacation). There are also local problems. Last Friday a message from SRI-STRIPE to SRI-CSL took 7 hours to be delivered, while TN and FTP between those two machines worked fine.

> From: David L. Edwards <DLE@SRI-STRIPE.ARPA>
> It is becoming increasingly apparent that there are serious network problems. There has been some discussion of this on the TCPIP forum. Network delays, failed connections, inability to make connections etc. are being reported by hosts all over the network.
>
> BillW has noticed that direct communications between SRI and SU are bad. In your case there are one or two gateways involved in addition to the IMPs. The mailer recently reported 750+ attempted connections with 670+ failures.

Old age? Software rot? Incompatible changes to net software? Saturation? Who knows. Stay tuned.

Search RISKS using swish-e

Report problems with the web pages to the maintainer

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 71

Tuesday, 30 September 1986

Contents

- Deliberate overrides?
  - Herb Lin
  - Alan M. Marcum
  - Eugene Miya
- "Friendly" missiles and computer error - more on the Exocet
  - Robert Stroud
- Re: Reliability, complexity, and confidence in SDI
  - Michal Young
- My understanding of "path" and "bathtub curve"
  - Bob Estell
- More artificial than intelligent? (Autokeywords)
  - Bob Estell
- A Viking lander query
  - PGN
- Note on ARPANET congestion
  - Nancy Cassidy
- Indeed, the network is getting old
  - Jonathan Young
- [MULTIPLE COPIES OF RISKS-3.70? Some of you were lucky enough to get 4 | copies, 2 after last reboot! Foonly Wizard David Poole strikes again? / Retries may also be due to net delays, timeouts in ACKS? Sorry. PGN] /
- Info on RISKS (comp.risks)

Deliberate overrides?

<LIN@XX.LCS.MIT.EDU>
Tue, 30 Sep 1986 00:58 EDT

From: Scott E. Preece

Deliberate Overrides - mechanical, even

Alan M. Marcum, Consulting <marcum@Sun.COM>
Tue, 30 Sep 86 09:56:41 PDT

Though perhaps not strictly computer related, I thought the following might be of interest to the Risks forum.

The Piper Arrow is a four-place, single-engine airplane with retractable landing gear. Piper has a wonderful airspeed switch in the landing gear system which will automatically lower the gear if the airspeed is too low. One side effect of this is that during a low-speed climb, the gear may drop (or might never come up during takeoff). Climbing with the gear down will seriously erode climb performance (up to 500 feet per minute, with max. climb around 1000 fpm), just when you want MAXIMUM climb performance!

To overcome this, Piper installed a "gear override" handle, which can be latched in the OVERRIDE position. Many, many Arrow pilots routinely take off with the override engaged, to ensure that the gear retract when the pilot wants them up.

Why did Piper install this mechanism? The reason most often cited is to help prevent gear-up landings. It is interesting that a number of Arrow pilots have landed gear-up, having forgotten to disengage the override after having gotten into the habit of depending on it.

I've flown retractable singles built by Piper, Cessna, Beech, and Mooney. Piper is the only one with the airspeed override. All manufacturers, including Piper, have a warning horn which sounds if power is reduced past some threshhold with the gear up, though.

What's the point? In my opinion, the automatic system increases pilot workload during a critical time (takeoff and initial climb). It's not something on which one should EVER depend: it might fail. It's prone to lower the gear at inopportune moments -- times a pilot would absolutely never lower the gear; times when having the gear down is seriously more dangerous than having the gear up (certain emergency situations, for example). And, you need the override.

Certainly, performance- or functionality-limiting devices can be useful. They must be thought through carefully, and considered as part of the whole, rather than as an isolated system.

Alan M. Marcum Sun Microsystems, Technical Consulting
...!{dual,decvax}!sun!nescorna!marcum Mountain View, California

---

Re: Deliberate overrides and multiple causes (blame)

Eugene Miya <eugene@AMES-NAS.ARPA>
30 Sep 1986 1330-PDT (Tuesday)

From: "Scott E. Preece" <preece%ccvaxa@GSWD-VMS.ARPA>
> /**** ccvaxa:fa.risks / LIN@XX.LCS.MI / 2:47 am Sep 29, 1986 ****/
> > From: Charles R. Fry <Chucko at GODZILLA.SCH.Symbolics.COM> [...]
> > From: LIN@XX.LCS.MIT.EDU [...]

Just a point of information: the Soviets just announced that they planned to get rid of reactor control rod overrides, and that one manual override at TMI accentuated the problem ("But overall system worked" summarizing the pro-nuclear viewpoint).

Is it possible to write a rule without exception?

>"You know, if I do what you've asked, the bomb is going to fall on the wing
> and probably strip off your starboard control surfaces."
>"Yes, I know, do it anyway."

Yes, I can see this happening, but it reminds me of the film Dark Star.

"Talk to the bomb . . . about phenomenology....."

--eugene miya, NASA Ames Research Center, eugene@ames-aurora.ARPA

Re: "Friendly" missiles and computer error - more on the Exocet

Robert Stroud <robert%cheviot.newcastle.ac.uk@Cs.Ucl.AC.UK>
Tue, 30 Sep 86 14:43:24 gmt

There is a very interesting BBC TV documentary in the Horizon series called "In the wake of HMS Sheffield" which is well worth seeing if you get the chance. It discusses the failures in technology during the Falklands war and the lessons which have been learnt from them, and includes interviews with participants on both sides.

Naturally the fate of HMS Sheffield features prominently, and the chronology given by Rob MacLachlan matches the program in most respects. However, I'm afraid it says nothing about the Exocet homing signal being friendly - I was specifically looking out for this. Instead, according to the documentary, the device which should have detected the homing signal is situated next to the satellite transmission device and was simply swamped by the signal from a telephone call to London in progress at the time - this backs up Peter's definitive account.

A couple of other points from the documentary are worth mentioning. Chaff was indeed effective in helping one ship avoid an Exocet (I forget which one) but it is by no means fool proof. The fuse needs to be set manually on deck and must be exact, taking into account lots of factors like wind direction, ship's course, distance from missile, etc. If you get it wrong, the distraction comes too early or too late. There was a nice piece of computer graphics showing the difference half a second could make - needless to say, they are working on an automatic fuse!

The Argentinian planes were able to avoid radar detection using a technique called "pecking the lobes". Basically they exploit the shape of the radar cone and the curvature of the earth by flying level until they detect a radar signal, then losing height and repeating the process. As Rob said, they only need to rise up high enough to be detected at the last minute when they fire the Exocets and turn for home - even this trace would only be visible very briefly on the radar display and could easily be missed. Thereafter the Exocets are silent until the last few seconds when they lock onto the target to make last minute course corrections.

This problem has been dealt with by building radar devices that can be used from helicopters several thousand feet up so they can see further over the horizon.

There was also a discussion about whether it would be feasible to install anti-missile weapons in cargo ships such as the Atlantic Conveyor (sunk twice by the Argentinians with Exocets who mistook it for one of the aircraft carriers). Apparently, installing a weapon would be possible, but to be effective it would need all the command & control computer systems as well to keep track of everything else that was going on, and that would not be feasible.

Robert Stroud, Computing Laboratory, University of Newcastle upon Tyne.

ARPA robert%cheviot.newcastle@cs.ucl.ac.uk (or ucl-cs.ARPA)
UUCP ...!ukc!cheviot!robert

Re: Reliability, complexity, and confidence in SDI software

Michal Young <young@ICSC.UCI.EDU>
Mon, 29 Sep 86 22:57:46 -0800

Bob's message, and some of the replies, seem to be using the term `path' in a sense I am unfamiliar with, since they refer to (large but) finite numbers of paths in software. If software contains loops, isn't the number of paths infinite? And therefore, after any finite amount of use, isn't the percentage of paths tested actually zero? If there is another commonly accepted meaning of `path' through a piece of software, please fill me in on it.

I have a similar problem with the term `state.' It seems to be used to refer to major states like `ready to run' and `running', whereas a fault may be sensitive to smaller-grain state like `i = 0 and j > 999'. It may be possible to design software to have a small number of major states, but the number of possible data+control states of any useful program is very large indeed.

> BOTTOM LINEs:
>
> 1. The curve for debugging software has a DOWNslope and length that is
> some function of the number of possible paths through the code.
> ...
> 3. Confidence builds as one approaches the 90% [or other arbitrary level]
> point in testing the number of possible paths.
>
> 4. The reason that we haven't built confidence in the past is that we've
> often run thousands of hours, without knowing either:
>
> a. how many paths got tested; or

> b. how many paths remained untested.

By the terminology I am familiar with, 3 is "never" and 4(b) is
"an infinite number" for every useful piece of software, always.

--Michal Young, UC Irvine, young@ics.uci.edu

My understanding of "path" and "bathtub curve"

"ESTELL ROBERT G" <estell@nwc-143b.ARPA>
30 Sep 86 09:04:00 PST

I don't claim to use "path" in a way that may be common in graduate courses in software engineering. My use is based on the highway map analog; e.g., there are many paths through the LA freeway system that one might take from Irvine to Mammoth on a ski weekend. One can drive any of the paths any number of times [loops?]; for lack of good all-way interchanges, some paths might not work well [design errors?]; because of temporary traffic congestion, some paths might be troublesome on some days [data sensitivity?].

I agree that software *is* sensitive to "minor" state conditions; e.g., loop counts of "zero" and "n+1" [where "n" was the intended limit] are notorious. I contend that it *should NOT* be; i.e., that proper design and testing can reduce such errors to a tolerable range. A goal of good software design is to construct "modules" whose internal states are insensitive to all legal arguments, and whose entry code screens out all illegal arguments; at least that's my personal understanding of one [of several] key benefits of "data hiding" and "defensive programming."

Another respondent disputed the "downslope" claim, because his experience was that the error rate degenerates to some constant level. Well, all the bathtubs I've seen do have bottoms. One can expect some non-zero number of bugs to persist; let's only hope that it's tolerably low - lower than during "alpha testing." Finally, if some "new release" goes badly sour [e.g., the "new" ARPANET s/w?] because it tries to "add on" [vice "design in?"] new features, maybe that's the equivalent to the "wear out" upslope in mechanical designs. That may be what we've seen with some older operating systems that tried to "add on" time sharing, security, or multi-processor logic.

Bob

More artificial than intelligent? (Autokeywords)

"ESTELL ROBERT G" <estell@nwc-143b.ARPA>
30 Sep 86 10:14:00 PST

Computer titles on documents are going to take over. Don't fight it.
It could be worse; they might have to be "bar coded."
Instead, just use "human" sub-titles; e.g.
ANTLERS, TREETOPS, MYSTERY; (or "Who Goosed the Moose?")

A Viking lander query

Peter G. Neumann <Neumann@CSL.SRI.COM>
Tue 30 Sep 86 20:26:06-PDT

Is there a RISKS reader who can report on what the Viking lander software really did? Was it used for landing? or just for communication and control of onboard equipment? "Working the first time" would be much more impressive if it were for landing, whereas the rest is more easily testable on the ground.

Note on ARPANET congestion

Nancy Cassidy <ncassidy@ccm.bbn.com>
Mon, 22 Sep 86 12:22:13 EDT

==
Report on Investigation Request #: IR86-0051-ARPANET-SY   Report #: 1
Date of Report: 9/22/86   Priority: 2
==
Reporting: open
==

IR Title: ARPANET congestion

Summary of Problem:

Another problem exists for users on MILNET who must access the BBNNET (especially users on DDN1 and DDN2). Currently, there is no gateway between the MILNET and BBNNET. Instead, traffic passes through the ARPANET to an ARPANET Gateway in order to reach the BBNNET. The critical congestion problems the ARPANET is experiencing causes TELNET and FTP connections to time out and mail messages from MILNET hosts to take up to 2-3 days to be delivered to BBNNET hosts.

One other result of network congestion is the Monitoring Center's ability to effectively monitor operations. The number of traps and status messages has increased proportionately to the severity of the congestion. This dramatic increase in network messages received by the MC consumes CPU space and slows down C/70 performance to the point where it affects monitoring and control of the network.

[Further reporting and recommendations truncated...]

Indeed, the network is getting old

Jonathan Young <young-jonathan@YALE.ARPA>

Tue, 30 Sep 86 12:59:47 edt

Here at Yale we have been aware of two problems: host tables are overflowing and mail is bouncing. Actually, we think that SENDMAIL connections (more often from BSD4.3 machines) are timing out and retrying. This has resulted in dozens of copies of certain messages. I enclose a copy of a message from our network administrator.

I'm very surprised that others haven't commented about the virtual unavailability of the ARPANET. On the other hand, Yale's connection is via a 9600 BAUD LINE to Harvard. Sigh.

Jonathan (YOUNG-JONATHAN@YALE.ARPA)

[Is that anywhere near the 50 YARD LINE? (rELIability!) PGN]

From: Morrow Long <long@YALE.ARPA>
To: department
Subject: ARPAnet mail problems

We began to see a large problem with repeating incoming arpanet mail messages in August (when cheops was still yale.arpa - the mail name host) - especially in the department bboard where a MIT site was flooding our newsgroups and bulletin boards with the mail internet bulletin board messages. After christening Yale-Eli as Yale.ARPA (a dedicated SMTP mail server) we have continued to experience the problem with repeating messages emanating from some hosts.

From statistics we have gathered on the problem we have noticed that many of the problem hosts are running 4.3bsd. Our problem may not be due to 4.3bsd TCP/IP (nor Sendmail/SMTP) but may be brought on by problems with arpanet congestion/delays wrecking session protocols.

To alleviate and eventually rectify the problem we have taken the following steps:

1. We have notified the administrators of the remote sites to remove the repeating messages from their spool queues.

2. We are tracing Sendmail/SMTP debugging messages to session logfiles to capture maximum information.

3. Luis has agreed to act as moderator for one of the most troublesome groups ('apollo@yale'), screening out duplicates before reposting them to the world.

4. A 'sweep' daemon has been created and installed on Eli to check for duplicate messages to bboards and mailing list in the mail queue and remove them for exact matches on Message-ID, sender and subject. At least one copy is always allowed through. Even this drastic program will allow repeat messages if they arrive outside of the queue sweep window for duplicates.

5. We will be investigating the 4.3bsd Unix sendmail program for incompatibilities with our SMTP servers.

H. Morrow Long, Computing Facility

[This is just the tip of the iceberg on reports and messages. The
TCP-IP BBOARD IS OVERFLOWING. All sorts of contributing factors are
being discussed. Dramatic increase in net traffic, total saturation of
the IMPs, hosts that stick with 4.2bsd instead of 4.3, weak gateways,
mail distributions to multiple users at the same site, etc. Who knows?
No one yet. The total collapse of the ARPANET on 27 Oct 1980 was only
for four hours or so, and has not happened since; this fiasco has been
going on for at least three weeks, and the network seems to be
rotting completely.

So, if you got this far in the issue, and are getting a private
copy of RISKS, please let me know if your site now has a BBOARD or
redistribution and you can live without a private copy of RISKS.
(Each time I suggest that I actually do get a few willing people.)
PGN]

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 72

Wednesday, 1 October 1986

Contents

- Viking Lander
  Nancy Leveson
- Deliberate override
  George Adams
- Overriding overrides
  Peter Ladkin
- A propos landing gear
  Peter Ladkin
- Paths in Testing
  Mark S. Day
- Confidence in software via fault expectations (RISKS-3.69)
  Darrel VanBuer
- Info on RISKS (comp.risks)

Viking Lander

<leveson@sei.cmu.edu>
Wed, 1 Oct 86 11:44:21 edt

I have spoken to a man who was involved in the building of the Viking Lander software. He told me that the on-board software consisted of three basic functions: terminal descent software, some transmission of data, and a sequence of operations while the lander was on the ground. The Honeywell computer had a memory of about 18,000 24-bit words. It was programmed in assembly language. There was an interrupt-driven executive program. What is interesting is his claim that there were several bugs in the software. There just were not any bugs that caused an abort of the mission or other serious consequences. The programs were, in fact, overloaded on the way to Mars because of discovery of some problems. This man claims that there was a great unwillingness to admit that there were bugs in the software and so some remain undocumented or difficult to find in the documentation. But the software was NOT bug-free.

There was also ground system software which was responsible for auxiliary computations such as pointing the antenna.

One of the reasons I have come into contact with this software is that I have been asked to participate in a fault tolerance experiment which will use the Viking lander as an example. In examining the software in the light of using it for such an experiment, Janet Dunham at RTI has found it not to be adequate in its present form because it is too small and straightforward. In fact, she is presently writing a specification of the terminal descent portion which adds complexity to the problem to make it more interesting and complex for the experiment (she is afraid that there just will not be enough errors made given the original problem). She says that the navigation portion of the terminal descent software (which is the most complex part) is only about 100 lines of Ada code.

The point to remember is that not all software is alike.
Small, straightforward problems with very little complexity in the logic (e.g., just a series of mathematical equations) may not say much about the reliability of large, complex systems. We know that scaling-up is a very serious problem in software engineering. In fact, it has been suggested that small vs. large software projects have very few similarities. It should also be noted that the avionics part of this relatively small and relatively simple software system cost $18,000,000 to build. Although the 18,000 words of memory were overloaded a few times, the amount of money spent per line of code was extremely high.

I worry when anecdotal evidence about one software project is used as "proof" about what will happen with general software projects. There just are too many independent variables and factors to do this with confidence. And, in fact, we do not even know for sure what the important variables are.

Nancy Leveson
Info. & Computer Science Dept.
University of California, Irvine

(arpanet address, for angry replies, is nancy@ics.uci.edu despite evidence above to the contrary)

Deliberate override

George Adams <gba@riacs.edu>
Wed, 1 Oct 86 10:18:43 pdt

Even automobiles might appropriately have overrides of automated controls, and even of automated safety systems. I have only read about the following automobile item and wish I had the opportunity to verify it, but it seems reasonable.

Consider the anti-lock braking systems now becoming more widely available in automobiles. The driver can apply a constant input to the brake pedal, but modulated braking forces are applied at the wheels so

that the wheels do not lock. Many have probably seen the ad on tele-
vision in which the car with anti-lock brakes sucessfully negotiates the
turn on wet pavement while coming to a rapid stop without skidding out of
control. Yet, perhaps such vehicles should have a switch to disable
anti-lock and allow conventional braking. Imaging trying to stop quickly
with anti-lock brakes on a gravel road.
Even if an incompetent driver forgot to enable the system on hard
pavement, performance would be no worse than now common. Without the
switch a competent driver might hit that cow instead of stopping in time.
Regarding aircraft, a report on the midair collision involving the
Aero Mexico flight said that the flight crew applied thrust reversers
after the collision. This seems like a creative response, and one that
might easily be disallowed in a more automated aircraft in which a
check for weight on extended landing gear was a prerequisite for thrust
reversal. While thrust reversal had no benefit for the Aero Mexico
flight itself, perhaps it reduced impact speed and consequently reduced
the extent of damage on the ground.
A vehicle and its operator are a system. By automating vehicle systems
we can adapt operator workload to better match the capabilities of human
beings and make it possible for an operator to do a better job. We can
also automate to limit operator options for coping with non-routine
situations and impede rapid operator override, thereby making a more
expert system and also a less generally capable one.

Overriding overrides

Peter Ladkin <ladkin@kestrel.ARPA>
Wed, 1 Oct 86 17:08:22 pdt

An example of a deliberate override that led to disaster:
An Eastern Airlines 727 crashed in Pennsylvania with considerable
loss of life, when the pilots were completing an approach in
instrument conditions (ground fog), 1000 feet lower than they
should have been at that stage.
They overrode the altitude alert system when it gave warning.

Peter Ladkin

A propos landing gear

Peter Ladkin <ladkin@kestrel.ARPA>
Wed, 1 Oct 86 16:55:14 pdt

Alan Marcum's comment on gear overrides in the Arrow reminded me
of a recent incident in my flying club (and his, too).
The Beech Duchess, a light training twin, has an override that
maintains the landing gear *down* while there is weight on the
wheels, ostensibly to prevent the pilot from retracting the
gear while on the ground (this is a problem that Beech has in
some of its airplanes, since they chose to use a non-standard
location for gear and flap switches, encouraging a pilot to

mistake one for the other).
Pilots can get into the habit of *retracting* the gear before takeoff, secure in the knowledge that it will remain down until weight is lifted off the wheels, whence it will commence retracting. This has the major advantages that it's one less thing to do during takeoff, allowing more concentration on flying, and the gear is retracted at the earliest possible moment, allowing maximum climb performance, which is important in case an engine fails at this critical stage.
Can anyone guess the disadvantage of this procedure yet?

Our club pilot, on his ATP check ride, with an FAA inspector aboard, suffered nosewheel collapse on take-off, and dinged the nose, and both props, necessitating an expensive repair and engine rebuild. Thankfully, all walked away unharmed.
It was a windy day.

It is popularly supposed that the premature retraction technique was used, and a gust of wind near rotation speed caused the weight to be lifted off the nosewheel. When the plane settled, the retraction had activated, and the lock had disengaged, allowing the weight to collapse the nosewheel.

Both pilots assure that the gear switch was in the down position, contrary to the popular supposition.

All gear systems in the aircraft were functioning normally when tested after the accident.

The relevance to Risks? The system is simple, and understood in its entirety by all competent users. The technique of premature retraction has advantages. It's not clear that a gedankenexperiment could predict the disadvantage.

Peter Ladkin

Paths in Testing

Mark S. Day <MDAY@XX.LCS.MIT.EDU>
Wed 1 Oct 86 11:57:32-EDT

"Paths" as used in the discussion of "path coverage" are probably intended to be what are called "basis paths." A piece of code with loops can indeed have an infinite number of paths, but every path is a linear combination of a much smaller set of paths. Testing that covers every basis path and also tests each loop using "engineer's induction" ("zero, one, two, three... good enough for me") is significantly better than random testing to "see what breaks" and much more feasible than trying to test all the combinations of basis paths.
The McCabe or cyclomatic complexity metric defines the number of basis paths through a piece of code; see

T.J. McCabe, "A Complexity Measure", IEEE Transactions on Software Engineering SE-2, 4 (Dec 1976) pp. 308-320.

A quick approximation of McCabe complexity is that straight-line code has a complexity of 1 (obviously, I guess) and most control statements (if-then, if-then-else, while, repeat, for...) each add 1 to the complexity. An n-way case statement adds n-1 paths to the "straight" path, so it adds n-1 to the complexity. This approximation only applies to code with no gotos.

The IEEE Computer Society puts out a tutorial volume called "Structured Testing" that includes the previously-cited paper and a number of other related articles, including a heuristic for using the McCabe complexity to select test paths.

--Mark

Re: Confidence in software via fault expectations (RISKS-3.69)

Darrel VanBuer <hplabs!sdcrdcf!darrelj@ucbvax.Berkeley.EDU>
Tue, 30 Sep 86 18:37:27 pdt

>From: Dave Benson <benson%wsu.csnet@CSNET-RELAY.ARPA>
>... Software ordinarily has no manufacturing defects and the
>usual way ordinary backups are done insures that most software does not
>wear out before it becomes obsolete.

The thing is software DOES wear out in the sense that it loses its ability to function because the world continues to change around it (maybe a bit because the pattern of bits does NOT wear out): e.g. operating systems which have gone psychotic because the number of bits used to represent a date because compatible hardware has continued to run far longer than designers of software and hardware anticipate (how many IBM-360 programs will correctly handle the fact that year 2100 is NOT a leap year, but still be running inside some emulation/automatic retranslation) or financial software unable to deal with 1000 fold inflation because all the numbers overflow...

Darrel J. Van Buer, PhD, System Development Corp., 2525 Colorado Ave
Santa Monica, CA 90406 (213)820-4111 x5449 / !sdcrdcf!darrelj
...{allegra,burdvax,cbosgd,hplabs,ihnp4,orstcs,sdcsvax,ucla-cs,akgua}

[This one is getting to be like the "YES, VIRGINIA, THERE IS A SANTA CLAUS" letter that used to appear each year in the Herald Tribune. But I keep reprinting the recurrences because some of you still don't believe it. There is no sanity clause. PGN]

Search RISKS using swish-e

Report problems with the web pages to the maintainer

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 73

Thursday, 2 October 1986

Contents

- Lessons from Viking Lander software
  Bob Estell
- Software wears out?
  Rob Austein
- Wrongful eviction through computer error
  Bill Janssen
- Deliberate override
  Herb Lin
  Ray Chen
- Re: Piper Arrow Gear Override
  Douglas Adams
- Undesirable breakins and causes
  Ian Davis
- Info on RISKS (comp.risks)

Lessons from Viking Lander software

"ESTELL ROBERT G" <estell@nwc-143b.ARPA>
2 Oct 86 12:18:00 PST

Angry replies? Never! To the contrary, Thank you, Nancy.
In the last few weeks in the "pages" of this journal, we have established:
1. Software cannot be perfect, because it is designed by fallible humans.
2. Hardware cannot be absolutely predictable, adding to the software woes.
3. Very concise programs, though not "perfect" nevertheless have sometimes
run well enough to be considered "successful" - even on a "first chance."
[Other than during simulated tests.]

It's time to move forward. I concur heartily with Nancy's point that what applies to small programs may not scale up to large programs. It's worth noting that in her last message, Nancy alluded to the Viking Lander as a "small" program, with "only" 18,000 words in assembly! How many of you agree that we might use orders of magnitude to distinguish sizes? e.g.,
IF xx,000 = small, then x,000 = tiny, and x00 = toy; and xxx,000 = large,

and x,000,000 = very large; etc.
Maybe it then follows that xx,000,000 is "possible?" [SDI size?]
Maybe not. Depends a LOT on HOW it's designed, coded, and tested.

I'd like those more knowledgable than I to address some of the following; maybe the software engineering journal is a better forum; maybe conclusions could be reprinted in RISKS.

a. What are the measures of "acceptability?"
[Analogy: In baseball, the 1.000 batting average [perfection] is never possible - except in trivial circumstances, then only with luck.
A century of experience says that .3xx is outstanding. Likewise, the 1.000 fielding average is "impossible." But anything less than .9xx is terrible.
What are similar failure rates for software? Where should we set our expectations? Where is the "point of diminishing returns?"
I'm not at all clear on just HOW to set these expectations.
[On a saturated UNIVAC 1110 a few years ago, we were suffering up to 3 crashes a day; we "engineered" that down to less than 3 per month.
Did that improvement make us just average, or much better?]

b. *IF* it's true that "small" programs can run "acceptably" albeit NOT perfectly, and that "very large" programs probably cannot (?), then why don't we get serious about building very large programs as orderly structures of small modules? (Where "small" may mean xx,000 lines.)
At the moment, it seems to me that we're caught on the horns of a dilemma of our own making; the "idealists" among us are saying that very large systems cannot be perfect, hence should not be pursued; the "realists" among us are saying that the present status of large, real-time systems is a disaster; the "analysts" among us are saying that there seems to be no good formula for success, yet; and the "pragmatists" among us are saying that we can make SOME worthwhile improvements in the status quo, and thus we should.
ALL FOUR VIEWPOINTS ARE "CORRECT."
Isn't it time now for all of us to start "groping" forward together?

Bob

Software wears out?

Rob Austein <SRA@XX.LCS.MIT.EDU>
Thu, 2 Oct 1986 00:47 EDT

The other sense in which software "wears out" is that people lose their ability to maintain it. I recently had to work on a mailer daemon that is about 15 years old. Fine code, possibly one of the best mailers ever written (certainly for its day), coded according to good programming practices -- for the early 1970s. I almost went nuts trying to modify it, people just don't think that way anymore (you know, labels every ten instructions, GOTOs everywhere...).

I was the person modifying the code because everybody else had better sense than to even try. At this point I can say with a fair amount of certainty that -nobody- really understands that program anymore, although the people who installed the various features (if still alive) usually remember having done so within a month or so of being asked. And this is a fairly small program compared to stuff done out in the Real World.

--Rob <BUG-COMSAT@MC.LCS.MIT.EDU>

Wrongful eviction through computer error

Bill Janssen <janssen@mcc.com>
Thu, 2 Oct 86 19:10:10 CDT

An interesting thing happened to me last month. I got home on the 5th of September to find an eviction notice on my living room floor. Something about not paying my rent. Well, I gathered up the checks and went over to the office. Turns out the problem was that I had already paid for October, as well as September, and the apartment management folks had just switched to a new computer system! There must have been a line in it something like

```
  if (last_month_paid_for != this_month
     AND day == trigger_day_for_eviction)

     issue_eviction_notice();
```

According to some of the office staff, 11 other people had already been in with similar complaints.

Bill Janssen, MCC Software Technology, 9430 Research Blvd, Austin, Texas 78759
UUCP: {ihnp4,seismo,harvard,gatech,pyramid}!ut-sally!im4u!milano!janssen

Deliberate override

<LIN@XX.LCS.MIT.EDU>
Thu, 2 Oct 1986 08:26 EDT

From: George Adams

Deliberate Overrides

Ray Chen <chen%gt-stratus%gatech.csnet@CSNET-RELAY.ARPA>
Wed, 1 Oct 86 02:37:57 EDT

Manual overrides are a nice idea, but chances are they'll be needed most during a sudden emergency when there isn't time to think about, much less trigger any kind of safety override.

How would you like to have to trigger a safety override while powering into a corner trying to avoid an accident? Ugh.

Personally, I don't see any way around it. Total control after all is just that -- the ability to specify exactly what the machine is going to do, even if it's beyond the normal performance envelope. Saftey restrictions on the other hand are designed to keep you from exceeding the performance envelope.

There's an inherent contradiction in the two objectives which can't be neatly resolved unless the safety system and the user/operator are always in perfect agreement on the limits of the performance envelope in every possible situation.

I think we have a trade-off here.

Ray Chen

uucp: ...!{akgua,decvax,hplabs,ihnp4,linus,seismo}!gatech!chen

CSNet: chen@GATech ARPA: chen%GATech.CSNET@CSNet-Relay.ARPA

Re: Piper Arrow Gear Override

Adams Douglas <crash!pnet01!adamsd@nosc.ARPA>
Thu, 2 Oct 86 08:59:20 PDT

Piper could also have installed the override system because some old lawsuit or other related to a gear-up landing would have caused their insurance rates to go through the roof if they didn't implement some kind of 'fix' that they could point to.

Incidentally, I don't advocate it, the problem of leaving the override on could be solved by having a flashing panel light next to the other two gear-status lights on the glareshield such as OVERRIDE ENGAGED. But that would simply add to the pilot's cockpit stimuli--which is never a good idea during takeoff.

Undesirable breakins and causes

Ian Davis <ijdavis%watdaisy.waterloo.edu@CSNET-RELAY.ARPA>
Thu, 2 Oct 86 17:32:33 edt

Has anyone suggested that hardware can also seriously undermine security? [YES] I tend to work from home and thus communicate via a modem, and have always logged off by merely switching from data to voice on my modem, which both drops the line, and hangs up the phone for me... unfortunately (and initially unbeknown to me) at least one of the modems that answers incoming calls from me (at a deliberately unspecified site) was hit by a current surge during a recent lightning storm, and now no longer drops the communication line to the CPU when I drop my line to it. This has disasterous consequences since the next caller to use this recieving modem finds themselves logged into my account with totally unrestricted access to the system. Fortunately most users are honest and promptly sign off, but the risks are very real.

The moral for those of you who are concerned, is that one should always log off from an operating system before dropping communication lines, and that one should log back on as soon as possible if the line is dropped accidentally.

Ian Davis

[We have been around that one several times now, although lightning hitting the modem is a new wrinkle! PGN]

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 74

Friday, 3 October 1986

Contents

- Opinions vs. Facts in RISKS Reports (re Aviation Accidents)
  Danny Cohen
- Mathematical checking of programs (quoting Tony Hoare)
  Niall Mansfield
- Risks of maintaining computer timestamps revisited [RISKS-3.57]
  Ian Davis
- Keyword indexing in automated catalogs
  Betsy Hanes Perry
- Re: Viking Landers -- correction
  Scott Preece
- Re: Confidence in software via fault expectations
  Scott Preece
- Overrides and tradeoffs
  Jerry Leichter
- Re: Deliberate overrides
  Brint Cooper
- Re: idiot-proof cars (risks-3.68)
  Col. G. L. Sicherman
- Info on RISKS (comp.risks)

Opinions vs. Facts in RISKS Reports (re Aviation Accidents)

<COHEN@B.ISI.EDU>
3 Oct 1986 09:18:27 PDT

Opinions vs. Facts in RISKS Reports (re Aviation Accidents)

Everyone is entitled to opinions and to facts. Keeping the two distinguishly separated is the basis of good reporting -- including the reports/contributions to RISKS.

The RISKS readers are best served by being able to tell one from the other, and to tell what is based on opinions/rumors and what on facts. Two examples follow.

In RISKS-3.27 Stephen Little reported about "one major accident in which the pilot followed the drill for a specific failure, as practiced on the simulator, only to crash because a critical common-mode feature of the system was neither understood, or incorporated in the simulation."

Since this is a very important evidence of "major accident" (with possible/probable loss of hundreds of lives) I tried to follow up on it and offered to pursue this report.

The best way to verify such a report is by a reference to the official NTSB (National Transportation Safety Board) accident investigation report. Therefore, I have volunteered to pursue this reference myself if anyone could give me details like the date (approximately), place (country, for example), or the make and type of the aircraft.

My plea for this information appeared in RISKS-3.34, on 8/9/1986.

In response, one RISKS reader provided me with a pointer to what he vaguely remembered to be such a case. After pursuing the original report we both found that the pilot (Capt. John Perkins, of United Airlines) claimed that [computer based] simulator training helped him and his crew to survive a windshear encounter (not the kind of story the RISKS community finds to be of interest).

(The long discussion about the F-16 does not relate to this topic since it was concentrated on what the simulator software should do and what the aircraft software should do, rather than on the fidelity of the simulator and on its training value).

If the original report about that computer-induced major accident is based on facts -- let's find them, we tried but did not succeed.
If it is based or rumors -- let's say so explicitly.

A more recent RISKS (3.72) has another report, this time by a pilot, Peter Ladkin, who also provides the place and the make and type of the aircraft (just as I asked for). His report says:

" An example of a deliberate override that led to disaster:
An Eastern Airlines 727 crashed in Pennsylvania with considerable loss of life, when the pilots were completing an approach in instrument conditions (ground fog), 1000 feet lower than they should have been at that stage.
They overrode the altitude alert system when it gave warning. "

I found it very interesting. The mention of the aircraft type and the location are helpful hints for pursuing such accidents.

However, I failed to locate any information about that "Eastern Airlines 727 [which] crashed in Pennsylvania".

I (and Eastern Airlines, too) know of only two losses of Eastern Airlines 727's -- neither in Pennsylvania. One in JFK to (windshear) and one in La Paz, Bolivia (flying into a mountain, in IFR conditions).

However, I know of the 9/11/1974 Eastern Airline crash of a DC-9 in Charlotte, North Carolina -- which, I guess, is what Peter Ladkin's report is about. This guess may be wrong.

I APOLOGIZE TO PETER LADKIN IF I DID NOT GUESS THE RIGHT ACCIDENT.

According to the NTSB accident report (NTSB-AAR-75-9) about the DC-9 in Charlotte: "The probable cause of the accident was the flightcrew's lack of altitude awareness at critical points during the approach due to poor cockpit discipline in that the crew did not follow predescribed procedure." [They were too low, and too fast.]

The report also mentions that "The flightcrew was engaged in conversations not pertinent to the operation of the aircraft. These conversations covered a number of subjects, from politics to used cars, and both crew members expressed strong views and mild aggravation concerning the subjects discussed. The Safety Board believes that these conversations were distractive and reflected a casual mood and a lax cockpit atmosphere, which continued throughout the reminder of the approach and which contributed to the accident."

What also contributed to the accident is that "the captain did not make the required callout at the FAF [Final Approach Fix], which should have included the altitude (above field elevation)". They also did not make other mandatory callouts.

Other possible contributing factors was a confusion between QNE and QFE altitudes (the former is above sea level, and the latter above the field elevation). [This may be the 1,000' confusion mentioned in Peter Ladkin's report.]

"The terrain warning alert sounded at 1,000 feet above the ground but was not heeded by the flightcrew" (which is typical to many airline pilots who regard this signal more of nuisance than a warning).

Question: What did Ladkin mean by "An example of a deliberate override
that led to disaster: ..... They overrode the altitude alert
system when it gave warning" ?

According to the NTSB they just did not pay attention to it. According to the Ladkin report they DELIBERATELY OVERRODE it, which implies explicit taking some positive action to override it. It is hard to substantiate this suggestion.

Not paying attention is not a "deliberate override" as promised in the first line of the Ladkin report, just as flying under VFR conditions into the ground is not "a deliberate override of the visual cues" -- it is a poor practice. (The only thing DELIBERATE in that cockpit was the discussion of used cars!)

Does this example contribute to the RISKS discussion about "deliberate override"?

In summary: Starting from wrong "facts" based on third hand vague recollections is not always the best way to develop theories.

Again, the RISKS readers are best served by more accurate reporting. They deserve it.

Danny Cohen.

Mathematical checking of programs (quoting Tony Hoare)

Niall Mansfield <MANSFIEL%DHDEMBL5.BITNET@WISCVM.WISC.EDU>
Thu 2 Oct 86 11:53:55 N

In "New Scientist", 18-Sep-86, C.A.R. Hoare discusses mathematical techniques for improving the reliability of programs, especially life-critical ones. The following somewhat arbitrary excerpts (quoted without permission) include some interesting ideas:

But computers are beginning to play an increasing role in "life-critical applications", situations where the correction of errors on discovery is not an acceptable option - for example, in control of industrial processes, nuclear reactors, weapons systems, oil rigs, aero engines and railway signalling. The engineers in charge of such projects are naturally worried about the correctness of the programs performing these tasks, and they have suggested several expedients for tackling the problem. Let me give some examples of four proposed methods.

The first method is the simplest. I illustrate it with a story. When Brunel's ship the SS Great Britain was launched into the River Thames, it made such a splash that several spectators on the opposite bank were drowned. Nowadays, engineers reduce the force of entry into the water by rope tethers which are designed to break at carefully calculated intervals.

When the first computer came into operation in the Mathematish Centrum in Amsterdam, one of the first tasks was to calculate the appropriate intervals and breaking strains of these tethers. In order to ensure the correctness of the program which did the calculations, the programmers were invited to watch the launching from the first row of the ceremonial viewing stand set up on the opposite bank. They accepted and they survived.

... [1.5 pages omitted]

I therefore suggest that we should explore an additional method, which promises to increase the reliability of programs. The same method has assisted the reliability of designs in other branches of engineering, namely the use of mathematics to calculate the parameters and check, the soundness of a design before passing it for construction and installation.

Alan Turing first made this suggestion some 40 years ago; it was put into practice, on occasion, by the other great pioneer of computing, John von Neumann. Shigeru Igarashi and Bob Floyd revived the idea some 20 years ago, providing the groundwork for a wide and deep research movement aimed at developing the relevant mathematical techniques. Wirth, Dijkstra, Jones, Gries and many others, (including me) have made significant contributions. Yet, as far as I know, no one has ever checked a single safety-critical program using the available mathematical methods. What is more, I have met several programmers and managers at various levels of a safety-critical project who have never even heard of the possibility that you can establish the total correctness of computer programs by the normal mathematical techniques of modelling, calculation and proof.

Such total ignorance would seem willful, and perhaps it is. People working on safety-critical projects carry a heavy responsibility. If they ever get to hear of a method which might lead to an improvement in reliability, they are obliged to investigate it in depth. This would give them no time to complete their current projects on schedule and within budget. I think that this is the reason why no industry and no profession has ever voluntarily and spontaneously developed or adopted an effective and relevant code of safe practice. Even voluntary codes are established only in the face of some kind of external pressure or threat, arising from public disquiet, fostered by journals and newspapers and taken up by politicians.

A mathematical proof is, technically, a completely reliable method of ensuring the correctness of programs, but this method could never be effective in practice unless it is accompanied by the appropriate attitudes and managerial techniques. These techniques are in fact based on the same ideas that have been used effectively in the past.

It is not practical or desirable to punish errors in programming by instant death. Nevertheless, programmers must stop regarding error as an inevitable feature of their daily lives. Like surgeons or airline pilots, they must feel a personal commitment to adopt techniques that eliminate error and to feel the appropriate shame and resolution to improve when they fail. In a safety-critical project, every failure should be investigated by an impartial enquiry, with powers to name the programmer responsible, and forbid that person any further employment on safety-critical work. In cases of proven negligence, criminal sanctions should not be ruled out. In other engineering disciplines, these measures have led to marked improvement in personal and professional responsibility, and in public safety. There is not reason why programmers should be granted further immunity...

... [1 page, to end of article, omitted]

Risks of maintaining computer timestamps revisited [RISKS-3.57]

Ian Davis <ijdavis%watdaisy.waterloo.edu@CSNET-RELAY.ARPA>
Wed, 1 Oct 86 17:47:29 edt

CP-6 has a further problem when first loaded that was encountered recently at Wilfrid Laurier University. A check is made to ensure that front end processors (FEP's) are up and running, but not that they contain the correct software... the consequence in W.L.U's case was that after loading version C01 for testing and then rebooting C00 software they left C01 software in the FEP's. Unfortunately, this resulted (for whatever reason) in disk record writes being interpreted as disk record deletes. The problem became apparent when using the editor which performs direct disk updates... but its severity was not at first appreciated... the system was brought down very rapidly when it was.... Ian Davis.

Keyword indexing in automated catalogs

Betsy Hanes Perry <betsy%dartmouth.edu@CSNET-RELAY.ARPA>
Wed, 1 Oct 86 10:40:39 edt

The recent notice about title-indexing (article titles must include all important article keywords in their first five words) struck a real chord in me. My current job is maintaining and updating Dartmouth College's automated card catalog.

We have a database of over 800,000 records, all completely free-text searchable (EVERY WORD in every record is indexed). We are beginning to suffer storage limitations, and are exploring our options. However, if we tried to suggest anything so restrictive as "five keywords per title", we'd have a revolution on our hands.

The instance cited seems to me to be a clear example of shaping the task to suit the tools at hand. Somebody out there ought to be ashamed of him/herself. At the very least, the notice explaining why articles' titles must be rewritten should have been

1. Extremely apologetic and
2. Should have given a time by which this temporary limitation would no longer apply.

As it stands, the system sounds as if it is going to be less useful than some of the available conventional journal indexes -- what incentive does this give for using it?

Tsk, tsk.

Re: Viking Landers -- correction

"Scott E. Preece" <preece%ccvaxa@GSWD-VMS.ARPA>
Thu, 2 Oct 86 09:33:19 cdt

> From: leveson@sei.cmu.edu
> Small, straightforward problems with very little complexity in the
> logic (e.g., just a series of mathematical equations) may not say much
> about the reliability of large, complex systems.

And there, of course, lies the heart of the structured programming
movement. You improve reliability by reducing the complexity of
program logic. You turn a large, complex system into a small,
straightforward system by building it in layers, each of which
makes use of primitives defined in the layer below.

The reason it may not be as effective as many have hoped is
that even simple, straightforward programs often turn out to
have bugs...

scott preece, gould/csd - urbana, uucp: ihnp4!uiucdcs!ccvaxa!preece

Re: Confidence in software via fault expectations

"Scott E. Preece" <preece%ccvaxa@GSWD-VMS.ARPA>
Thu, 2 Oct 86 09:25:04 cdt

> From: hplabs!sdcrdcf!darrelj@ucbvax.Berkeley.EDU (Darrel VanBuer)

> The thing is software DOES wear out in the sense that it loses its
> ability to function because the world continues to change around it...

That's like saying "People do live forever in the sense that some of their
atoms linger." The sense you depend on is not in the words you use.

"Becoming obsolete" is NOT the same thing as "wearing out." The word "wear"
is in there for a reason. Software does not suffer wear (though storage
media do). The only exception I can think of would be demonstration
packages that self-destruct after a set number of uses.

Words are important; if you smear their meaning, you lose the ability to say
exactly what you mean. This is a risk the computing profession has
contributed to disproportionately.

scott preece

Overrides and tradeoffs

<LEICHTER-JERRY@YALE.ARPA>
3 OCT 1986 13:26:54 EST

The recent discussions on manual overrides for airplane landing gear and car
brakes have all been ignoring a fundamental issue: To compute the expected
cost/risk of having/not having an automated system, you need more than just a
few gedanken experiments; you need some estimates of the probabilities of
various situations, and, in each of those situations, the expected costs of
using or not using the automatic systems.

Here's a simple, well-known example: Some people claim they don't wear seat
belts because, in an accident, they might be trapped in a burning car, or one
sinking into a lake. Is this a valid objection? Certainly; it COULD happen.

But the reality is that such accidents are extremely rare, while accidents in which seat belts contribute positively are quite common. So, on balance, the best you can do is wear seat belts. Of course, if you are in some very special situation - doing a stunt that involves driving a car slowly across a narrow, swaying bridge over a lake, for example - the general statistics fail and you might properly come to a different conclusion.

In the United States, how many people regularly drive on gravel roads? Perhaps for those relatively few who do, an override for the automatic brake system, or even a car WITHOUT such a system might make sense. Perhaps the costs for all those people who almost never drive on gravel roads can be shown to be trivial. There certainly ARE costs; every additional part adds cost, weight, something that can break; plus, there's another decision the driver might not want to be burdened with. And there are "external" costs: An uncontrolled, skidding car could easily injure someone besides the driver who chose to override the ABS.

Accidents in general are fairly low-probability events. As such, they have to be reasoned about carefully - our intuitions on such events are usually based on too little data to be worth much. Also, since we have little direct experience, we are more likely to let emotional factors color our thinking. The thought of being trapped in a burning or sinking car is very disturbing to most people, so they weight such accidents much more heavily than their actual probability of occurrence merits.

It's also worth remembering another interesting statistic (I wish I knew a reference): When asked, something like 80% of American male drivers assert that their driving abilities are "above average". Given such a population of users, there are risks in providing overrides of safety systems.

-- Jerry

Re: Deliberate overrides

Brint Cooper <abc@BRL.ARPA>
Fri, 3 Oct 86 13:53:54 EDT

> ..... Yet, perhaps such vehicles should have a switch to disable
> anti-lock and allow conventional braking. Imaging trying to stop quickly
> with anti-lock brakes on a gravel road...

But the whole point of anti-lock brakes is to avoid skidding when traction is lost. If the vehicle skids, it'll hit the cow. Overrides, as has been said before, allow incompetent operators to substitute their opinions for facts.

Brint

Re: idiot-proof cars (risks-3.68)

"Col. G. L. Sicherman" <colonel%buffalo.csnet@CSNET-RELAY.ARPA>

Mon, 29 Sep 86 09:15:13 EDT

Chuck Fry's argument for override provisions in automated controls on cars makes a lot of sense. Frankly, though, I'd rather see as few new automatic controls as we can manage with. I live in the Buffalo area--heavy industry with cobwebs on it--and people here are driving cars that ought to have been junked last year.

Airplanes get first-class maintenance, or at least second-class. With cars it's different; when something breaks, many people just can't afford to have it fixed. The simpler a car's design, the longer a poor man can keep it running safely.

Maybe I'm being cynical, but I believe that so simple an improvement as putting brake lights on rear windshields will prevent far more accidents than any amount of intermediary computerization.

[Since deregulation, you might be surprised that the airlines like everyone else believe in cutting expenses to the bone. Maintenance may or may not be what it was. I have seen several reports that it is not, although it is certainly nowhere near so bad as with autos. PGN]

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 75

Saturday, 4 October 1986

Contents

- re: Estell on Viking (RISKS-3.73)
  - David Parnas
  - Dave Benson
- Software becomes obsolete, but does not wear out
  - Dave Benson
- The fallacy of independence
  - Dave Benson
- Re: Paths in Testing (RISKS-3:72)
  - Chuck Youman
  - Mark Day
- Mathematical checking of programs (quoting Tony Hoare)
  - Henry Spencer
- Info on RISKS (comp.risks)

---

re: Estell on Viking (RISKS-3.73)

<parnas%qucis.BITNET@WISCVM.WISC.EDU>
Fri, 3 Oct 86 15:33:03 EDT

Robert Estell's contribution perpetuates two serious myths about the discussion on Viking and other software.

(1) That any of the discussants is expecting perfection.

Perfectionists do not use the net. In fact, the only computer scientist I know who could be called a perfectionist does not use computers. Most of us know that computer systems, like other human artifacts, will never be perfect. Our concern is with establishing confidence that the system is free of unacceptable or catastrophic errors. This we can do with many other engineering products. Only software products regularly carry disclaimers instead of limited warranties. That is not because they are the only products that are imperfect. It is because we have so little confidence that they are free of catastrophic flaws.

(2) That size is a good measure of the difficulty of a problem.

There are big programs solving dull but easy problems. Small programs occasionally solve very hard problems. The size and irregularity of the problem state space, and how well we know that state space determine, in large part the complexity of the problem. The size of the program is often determined by the simplicity of the programmer.

In spite of Nancy's help, we don't know much from this forum about what the Viking software actually did. It seems clear that most of the software could have been, and was, used before the flight. Whether the descent software could have been used depends on what it did. At 100 lines one would expect that it did not do much.

We all know that programs can work acceptably well. We use them and accept what they do. We also know that failures are not catastrophic and that these programs failed many times before they became reliable enough to be useful. If we had been in a situation in which those failures were unacceptable we would have found another way to solve the real problem.

Viking Lander, once again.

Dave Benson <benson%wsu.csnet@CSNET-RELAY.ARPA>
Fri, 3 Oct 86 17:57:52 pdt

I repeat some quotations from Bonnie A. Claussen's paper:
   The unprecented success of the Viking mission was due in part to
      the ability of the flight software to operate in an AUTONOMOUS
      and ERROR FREE manner. ... Upon separation from the Orbiter the
      Viking Lander, under AUTONOMOUS software control, deorbits, enters
      the Martian atmosphere, and performs a soft landing on the surface.
      [CAPS added for emphasis.]
Since the up-link was only capable of 4 bits/sec and the light-speed signal requires about 14 minutes for a round-trip to Mars, manifestly the software carried out these control functions without human assistance.

>I worry when anecdotal evidence about one software project is used as
>"proof" about what will happen with general software projects.
>   Nancy Leveson

I concur. But the Viking Lander experience does give a compelling example that autonomous software can be made to work under certain circumstances. Thus a claim that <

Software becomes obsolete, but does not wear out

Dave Benson <benson%wsu.csnet@CSNET-RELAY.ARPA>

Fri, 3 Oct 86 18:33:12 pdt

ob'so.lete. Abbr. obs. Of a type or fashion no longer current; out of
date; as, an obsolete machine.

ob'so.les''cent. Going out of use; becoming obsolete.

wear v.t. ... 6. To use up by wearing (sense 1); as, to wear out a dress;
hence, to consume or cause to deteriorate by use, esp. personal use;
as, the lugage is worn. 7. To impair, waste, or diminish, by continual
attrition, scraping, or the like; as, the rocks are worn by water;
hence, to exhaust or lessen the strength of; fatigue; weary; use up;
as, to be worn with desease. 8. To cause or make by friction or wasting;
as, to wear a channel or hole.
wear v.i. ... 4. To be wasted, consumed, or diminished, by use; to
suffer injury, loss, or extinction, by use or time;-- often with
<out>, <off>, <on>, etc.; as the day has worn on.

Software, like any artifact, becomes obsolete over time. The changing
informational environment about the software drives it to obsolesence.
It becomes unmaintainable, not from wear, but because the expertise
required has become dissipated. Recall that nobody knows how to
make a Stradivarius violin anymore, either.

I agree with the causes of software obsolescence, but strongly recommend that
we use the customary meanings of words in the dictionary so that
we understand one-another and so that non-software-types can somewhat
understand us as well. Thus: software may become obsolete from many
causes, some of which are understood. But software ordinarily does not
wear out and never, never rots. [...]

There is a reason for precise technical terms. In other disciplines words
are coined, just to avoid the overloading and potential resultant
misunderstanding. I recommend that we attempt this, but suggest looking
in the dictionary first.

The fallacy of independence

Dave Benson <benson%wsu.csnet@CSNET-RELAY.ARPA>
Sat, 4 Oct 86 22:21:39 pdt

A RISKS contribution suggests that since we can engineer good 100,000
statement software, the means to make good 1,000,000 statement software
is to produce 10 smaller packages and hook these 10 together.
Such a claim makes the assumption that the informational
environment of the total software is such that the total software system
can be decomposed into 10 nearly independent parts, which communicate with
one another along well-understood interfaces.
The key is the claim that the interfaces are well-understood.
Software is an example of an extremely complex artifact, a class of artifacts
which we understand poorly--for otherwise we wouldn't call them complex.
In smaller programs we repeatedly see that the interfaces are not

well-understood until the program is available for experimentation. Even then, our everyday experiences with software demonstrate again and again that what we had assumed about the program behavior does not match the reality of actual experience. Thus we discover that the interfaces are not well-understood.

Example: Virtual storage managers in operating systems provide a superficially simple interface to the hardware and the rest of the operating system. The interface to the user program is the essense of simplicity--complete transparency. Now the earliest virtual storage managers were the essense of simplicity. So nothing could go wrong, right? Wrong. The interaction of user virtual storage requests, the operating system scheduler, and the virtual storage manager led to thrashing--slowing performance to a crawl, at best. Upon OBSERVING this phenomenon, theories were developed and better, more complex, algorithms were installed. But this phenomenon was not predicted a priori.

The essential point is that even the cleanest design may fail in actual engineering practice until it is tried in the fully operational environment for which it was intended. In software engineering we only have confidence in a design if it is similar to a previous, successful design. But that is just like any other engineering practice. The intuition and insight of a Roebling (Brooklyn Bridge, 1883) is rare in any engineering field. Most of us are good copiers, making local improvements to a design already shown to be successful.

The corollary is that it is wrong to assume the near-independence of components until this near-independence has been abundantly shown in practice and theory.

Example: The division of the frontend of a compiler into lexical and syntactic parsing components which interact in well-understood ways has an excellent underlying theory and works well in practice. Thus it is common to teach this practice and theory, since post-facto it is a workable engineering design of nearly-independent components.

By all means color me realist. Also color me existentialist. What works is that which works, not what we might hope or dream or imagine works. The near-independence of software components is an aspect which is proved in practice to be a near-independence of components. As there is no "software decomposition theorem" which provides a general framework for that elusive quality, near-independence, we cannot assume that 10 good parts will actually form a cohesive, practical reliable whole. In each separate design, then, the value of the whole system can only be demonstrated by the use of the whole system.

Thus I claim it is a fallacy to assert independence, or even near-independence, for any division of the work within a system until this has been conclusively demonstrated. I further claim, with ample historical precedent, that the reliability of a system is only poorly correlated with the reliability of its parts. Without a specific design one can say nothing in general.

Re: Paths in Testing (RISKS-3:72)

Chuck Youman <m14817@mitre.ARPA>
Fri, 03 Oct 86 16:46:13 -0500

A comment on basis paths.

There was a paper on "Evaluating Software Testing Strategies" presented by Richard Selby at the 9th Annual NASA Goddard Software Engineering Workshop that compared the strategies of code reading, functional testing, and structural testing in three aspects of software testing. One of the conclusions I recall is that structural testing was not as effective as the other two methods at detecting omission faults and control faults.

The conference proceedings are report SEL-84-004 and can be obtained from Frank E. McGarry, Code 552, NASA/GSFC, Greenbelt, MD 20771.

Charles Youman (youman@mitre.arpa)

Re: Paths in Testing (RISKS-3:72)

Mark S. Day <MDAY@XX.LCS.MIT.EDU>
Sat 4 Oct 86 14:03:24-EDT

It's reasonably well known that structural (path-based) testing is poor at detecting faults of omission. Correspondingly, functional testing is poor at detecting faults on "extra" paths that are present in the implementation (for optimization of common cases, for example) but are not "visible" in a functional spec of the module. The conclusion to draw is that proper testing requires a combination of "external" testing (treating the module as a black box and examining its input/output structure) and "internal" testing (examining the contents of the module).

--Mark

Mathematical checking of programs (quoting Tony Hoare)

<decvax!utzoo!henry@ucbvax.Berkeley.EDU>
Sat, 4 Oct 86 21:12:13 edt

I agree with much of the quoted discussion from Hoare, including the obvious desirability of rather heavier use of mathematical analysis of safety-critical programs. I do have one quibble with some of his comments, though:

> ... never even heard of the possibility that you can establish
> the total correctness of computer programs by the normal mathematical
> techniques of modelling, calculation and proof. ...
> A mathematical proof is, technically, a completely reliable method of
> ensuring the correctness of programs, but this method could never be
> effective in practice unless it is accompanied by the appropriate attitudes
> and managerial techniques. ...

I think talk of "total correctness" and "complete reliability" shows excess enthusiasm rather than realistic appreciation of the situation. Considering the number of errors that have been found in the small programs used as published examples of "proven correctness", wariness is indicated. Another cautionary tale is the current debate about the validity of the Rourke/Rego proof of the Poincare conjecture. As I understand it -- it's not an area I know much about -- the proof is long, complex, and sketchy, and nobody is sure whether or not to believe it. And this is a case where the specs for the problem are very simple and obviously "right". Mathematical proof has its own feet of clay. If one defines "effective in practice" to imply complete confidence in the results, then I would not fly on an airliner whose flight-control software was written by a team making such claims. Complete confidence in provably fallible techniques worsens risks rather than reducing them.

(The apocryphal comment of the aeronautical structure engineer looking at his competitor's aircraft: "Fly in it? I wouldn't even walk under it!")

On the other hand, if one defines "effective in practice" to mean "useful in finding errors, and valuable in increasing one's confidence of their absence", I wholeheartedly agree. One should not throw out the baby with the bathwater. If one sets aside the arrogant propaganda of the proof-of-correctness faction, there is much of value there. To borrow from the theme of a PhD thesis here some years ago, proving programs INcorrect is much easier than proving them correct, and is very useful even if it isn't the Nirvana of "total correctness". The mental discipline imposed on program creation (defining loop invariants, etc.) is also important.

Henry Spencer @ U of Toronto Zoology
{allegra,ihnp4,decvax,pyramid}!utzoo!henry

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 76

Sunday, 5 September 1986

Contents

- Obsolescence vs wearing out (RISKS-3.75)
  Jerome H. Saltzer
- Cars, computers and unexpected interactions
  Mike McLaughlin
- Re: Mathematical checking of programs (quoting Tony Hoare)
  Matthew Wiener
- "Total correctness", "complete reliability" (RISKS-3.75)
  Bard Bloom
- Info on RISKS (comp.risks)

Obsolescence vs wearing out (RISKS-3.75)

Jerome H. Saltzer <Saltzer@ATHENA.MIT.EDU>
Sun, 5 Oct 86 14:14:16 EDT

Dave Benson nicely identifies a distinction between becoming obsolete and wearing out, and argues that only the former applies to software.

There is another effect that isn't exactly captured by the words "to become obsolete." A high quality piece of software, carefully designed and debugged by an expert, is turned over to a less-skillful operations team which installs it, runs it--and adds minor field modifications. As time goes on users of the software notice that it is no longer bug-free, because the less-skillful modifiers have been screwing it up. It isn't appropriate to say that the software became obsolete; if it hadn't been tinkered with then the term obsolete might apply.

Since most software does get modified to meet changing conditions of use, and often those modifications are not done by the original implementation team, this effect is quite common. If the effect goes on long enough, it may be necessary to commission a new implementation, almost as if the original implementation had worn out.

Some people have in mind the impairment and diminished usability
caused by this effect when they use words like "wears out" or "rots".
I guess we need a plain English word for it so that neophytes won't
think that computers that haven't been oiled properly rub too hard
on the bits.
Jerry

Cars, computers and unexpected interactions

Mike McLaughlin <mikemcl@nrl-csr>
Sun, 5 Oct 86 16:33:33 edt

1. I have a 1983 Ford with "Cruise Control"
2. I have had a CB in it from the day it was picked up (7/3/83)
until the day the CB was stolen (10/2/86). No problems.
3. I put a new, more sophisticated CB in it on 10/4. New CB has an
SWR (Standing Wave Ratio) meter, and an "Antenna Warning" light. Both
intended to help tune antenna system, and ensure crummy antenna connections
don't cause loss of signal strength - or excessive reflection of trans-
mitted signal.
4. SWR of 1.0 is perfect, and impossible. SWR of 1.5 is good. SWR of
2.0 is poor. SWR > 3.0 UNSAT!
5. New CB installed with only trivial cursing and sweating. Tuned up just
fine. Car drove fine (as before).
6. Rains came. SWR > 3.0. Probable cause bad antenna connections/cable,
getting soaked. Cruise control acted up. Wonder why?
7. Car baked in sun. SWR < 2.0. Cruise control OK.
8. This morning, car wet from heavy dew. SWR > 3.0. Cruise control cuts
out when microphone is keyed. Every time.
9. Car dries out, SWR < 2.0, Cruise control not affected.
10. SWR ratio must have varied with moisture on old set, same as new.
Never had problem before... but did re-route the power cables to new set,
more "neatly" than before, i.e., more jammed up behind instrument panel.
Conclusion: New CB/re-routed wiring somehow interacts with "Cruise Control"
micro, causing it to kick out when SWR is high. At least it "fails safe."
N.B.: I don't usually drive in rain with cruise control on, but do use it w
whenever safe to do so - saves gas on level-ish interstates. - Mike

Re: Mathematical checking of programs (quoting Tony Hoare)

Matthew P Wiener <weemba@brahms.Berkeley.EDU>
Sun, 5 Oct 86 16:00:39 pdt

In response to utzoo!henry (Henry Spencer):

<> A mathematical proof is, technically, a completely reliable method of
<> ensuring the correctness of programs, ... [from a Hoare quotation]

>I think talk of "total correctness" and "complete reliability" shows excess
>enthusiasm rather than realistic appreciation of the situation.

Agreed.

Henry then compares this notion of proof with the Rourke-Rego "proof" of the Poincare conjecture, whose status currently is unknown. And as Henry says, in mathematics
>the specs for the problem are very simple and obviously "right".

I must take exception to this comparison.

Mathematics, believe it or not, works under the Hundredth Monkey Phenomenon. Programs do not.

Let me explain.

Proofs in mathematics (at least at the cutting edge) deal with inherently complicated mentally defined objects. It takes a while to get your mind in sync with whatever it is you are studying. Details and (not always elementary) claims are left to the reader. The field, already huge beyond comprehension, would sink under its own weight otherwise.

New and difficult proofs, like that of Rourke-Rego, take their time to sink into the mathematical community's collective consciousness. But once they do, a new level of confidence and ability is reached, and the proofs become accessible.

The above is not possible with programs. At some point, every detail must be given, somewhere. There is no reason why a proof-checker could not be used to check for correctness, matching pre-and-post assertions with each statement.

So, where do "proven" programs fall down?

First, there really are the incorrect proofs. But I believe this can be cured. (Of course, relying on a proof-checker could be risky if *that* program has bugs. But surely that is a low enough operation to get right. [And now a new {recursive} nightmare comes to mind.])

Second, compilers and hardware do not always match the programmer's intent. Hidden pointer nonsense, erroneous implementations of mathematical functions, silent truncation of overflows, etc. cannot be checked for unless the programmer is aware of such glitches.

Third, the outside world need not match the programmer's intent either. The beginning assignment of input, and the final interpretation of output is outside the program's proof's scope. GIGO, as we all know.

Fourth, the theoretical process being used may be incorrect or just inappropriate in a particular situation. One can give your numerical analysis routines a proof that they do what is wanted, and build your aircraft or nuclear reactor or what have you with a new false confidence, despite the fact that the case at hand is subject to numerical instability or similar problems.

So in summary, a program and its proof are meaningful relative to each other, and nothing else. I would hate to think of the consequences if someone forgot this when implementing SDI, say.

ucbvax!brahms!weemba   Matthew P Wiener/UCB Math Dept/Berkeley CA 94720

"Total correctness", "complete reliability" (RISKS-3.75)

Bard Bloom <bard@THEORY.LCS.MIT.EDU>
Sun, 5 Oct 86 10:48:52 edt

>From: decvax!utzoo!henry@ucbvax.Berkeley.EDU
>I think talk of "total correctness" and "complete reliability" shows excess
>enthusiasm rather than realistic appreciation of the situation...

"Total correctness", at least, is a technical term in program verification. "Partial correctness" means that the program does the correct thing iff it terminates (i.e., the program that never terminates is partially correct). Total correctness is, partial correctness together with termination. All of these terms really mean "meets the mathematical specification".

>Another cautionary tale is the current debate about the validity of the
>Rourke/Rego proof of the Poincare conjecture. As I understand it -- it's
>not an area I know much about -- the proof is long, complex, and sketchy,
>and nobody is sure whether or not to believe it. And this is a case
>where the specs for the problem are very simple and obviously "right".

The proofs of program correctness are (supposed to be) checked by machines. There's been a lot of work done (and even a little success, I think) in getting proof techniques that can be checked automatically, and even ways of getting the machine to do a lot of the drudgework in converting a human-style proof into a machine one. Of course, you have to check the proof-checker...

As I understand the area of correctness proofs, there are two major problems:

1) Program specifications (especially complicated ones) rarely specify what you want the program to do. Not a whole lot program verification can do about this.

2) It is very hard to prove a program correct. Loop invariants, for example, are rather hard to come up with. Once you have the proof, it's easy to check.

> To borrow from the theme of a PhD thesis here some years ago, proving
> programs INcorrect is much easier than proving them correct,

I agree. The rumor around here is, the best use of program-proving techniques is in finding bugs.

-- Bard Bloom

Search RISKS using swish-e

Report problems with the web pages to the maintainer

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 77

Wednesday, 8 October 1986

Contents

- Evaluating software risks
  Brian Randell
- Misapplication of hardware reliability models
  Nancy Leveson
- Deliberate overrides?
  Mark Brader
  Ephraim
- Trusting-infallible-machines Stonehenge anecdote
  Mark Brader
- [More Aviation Hearsay?]
  C Lewis
- Info on RISKS (comp.risks)

Evaluating software risks

Brian Randell <brian%cheviot.newcastle.ac.uk@Cs.Ucl.AC.UK>
Mon, 6 Oct 86 18:59:47 gmt

In the article by Nancy Leveson in RISKS 3.72, she mentioned that a task based on the Viking Landing software was going to be used as the basis for some experiments on concerning fault tolerant software. There have in fact been a number of carefully cntrolled experiments aimed at assessing the possible cost-effectiveness of fault tolerance in software.
I then read the Tony Hoare quote in RISKS 3.74, bemoaning the fact that formal verification had not been used in any safety-critical software. Offhand, I do not know of any similar controlled experiments being performed on the cost-effectiveness of formal verification. Indeed, it strikes me that it would be very interesting if the planned experiments that Nancy refers to were to cover various verification and testing, as well as, fault tolerance experiments. Ideally risks should be quantified, so claimed remedies should be the subject of experimental evaluation, as well as eloquent pleading.

Brian Randell - Computing Laboratory, University of Newcastle upon Tyne

ARPA : brian%cheviot.newcastle@ucl-cs.arpa
UUCP : <UK>!ukc!cheviot!brian
JANET : brian@uk.ac.newcastle.cheviot

Misapplication of hardware reliability models

<leveson@sei.cmu.edu>
Mon, 6 Oct 86 11:39:34 edt

There has been some discussion on Risks lately about the application of hardware reliability models to software. The purpose of such models is to make predictions. The accuracy of a prediction based upon a mathematical model depends on whether the assumptions of the underlying model fit the situation in which it is being applied. Hardware reliability models make such assumptions as:

1) component failures will occur independently in independent replicated units
2) the behavior of a physical component can be predicted from data gathered from observations of other components that are assumed to be similar.
3) the design of the system is free from faults.

None of these seem to apply to software. Attempting to come up with some strange meaning of "wear out" so that the models can be applied to software is begging the question. We know that "wear out" AS IS MEANT IN THESE MODELS does not apply to software. Therefore, the results of applying the models to software may be inaccurate. The burden of proof is in showing that the assumptions apply as originally conceived in the models. As an example, trying to fit software to "bathtub curve" models (which were built by observing hardware) would seem to be a less fruitful line of endeavor than attempting to build models from what we observe about software.

Nancy Leveson
Info. & Computer Science
University of California, Irvine

Deliberate overrides?

<decvax!utzoo!dciem!msb@ucbvax.Berkeley.EDU>
Wed, 8 Oct 86 04:32:03 edt

City buses on this continent generally have rear or center exit doors interlocked with the brakes, so that once the exit doors are opened, the bus cannot be moved until they are closed. An excellent safety feature, yes? You'd never move a bus while someone might be getting off...

Well, one day a few years ago, on the Toronto transit system, the exit doors of a bus popped open spontaneously due to a malfunction in their control system, and stayed open. The bus was on a level crossing, and was full, and a few seconds later the barriers started lowering as a train approached. The collision was frightful.

In the investigation it turned out that the buses were fitted with a control to override the interlock, but it was in a concealed location (for maintenance access only) and drivers were not trained in its use. Needless to say this was promptly changed.

On the other hand, I could also cite several instances in the well-documented history of British railway accident investigations where both drivers and signalmen* were provided with overrides to be used only in case of equipment malfunction, and did not believe their equipment, and used the overrides to cause accidents.

*They WERE all men in those days. I don't know what the modern word is.

The moral seems to be that overrides are indeed a good thing to have, but you have to be very sure that the user knows when to use them. And if the engineer or programmer isn't the one training the users, this can be rather difficult.

By the way, those reading this somewhere else than on Usenet may be interested to know that people who use an interface called Pnews to post Usenet articles are asked:

  This program posts news to many hundreds of machines throughout the world.
  Are you absolutely sure that you want to do this? [ny]

This comes up before the message is entered; afterwards, the question

  Send, abort, edit, or list?

is asked, so the initial question is not the only chance to abort. In effect, the extra initial confirmation asks users to override a safety feature on every normal invocation of the program. Is this useful?
   [ANSWER TO MARK PLEASE, NOT RISKS ON THIS QUESTION.]
Mark Brader, utzoo!dciem!msb

Re: Overrides and tradeoffs (Risks 3.74)

<decvax!wanginst!wang!ephraim@ucbvax.Berkeley.EDU>
Tue, 7 Oct 86 20:20:56 edt

In Risks 3.74, Jerry Leichter writes:
>Accidents in general are fairly low-probability events. As such, they have to
>be reasoned about carefully - our intuitions on such events are usually based
>on too little data to be worth much. Also, since we have little direct expe-
>rience, we are more likely to let emotional factors color our thinking. The

>thought of being trapped in a burning or sinking car is very disturbing to
>most people, so they weight such accidents much more heavily than their actual
>probability of occurrence merits.

An interesting article on this topic (perception of risk) appeared in Scientific American a few years back. To summarize, small non-zero risks have much more emotional weight than they "deserve" (statistically, that is). Large variations in the middle of the scale have less effect than they deserve. Memory fails me on how risk at the other end of the scale (near certainty) is perceived.

Personally, I find the thought of being sent through the windshield at least as disturbing as (and much more likely than) being trapped in a burning car.

Trusting-infallible-machines Stonehenge anecdote

<decvax!utzoo!dciem!msb@ucbvax.Berkeley.EDU>
Wed, 8 Oct 86 05:14:25 edt

In the 1973 book "Beyond Stonehenge", Gerald S. Hawkins is telling about the digitization of the layout of Stonehenge from new aerial photos ...

> Back at the laboratory, two pictures, red and green, are projected. The operator looks through special glasses. A miniature Stonehenge sits there in the machine, three-dimensional, vividly real. A small white spot moves in the machine, controlled by hand dials. It can be moved along the ground; up ... down ... The machine reads height of stone or height of ground above datum. The method is accurate, absolute, unambiguous, mechanically final. The details are safely left with the engineer.
>
> When I saw the first photogrammetic plan I was puzzled. The number of stones was wrong. There was an extra stone mapped in the bluestone horseshoe.
>
> I raised the question with Mr. Herschel. The engineers put the film back in the infallible machine and redid the measurements.
>
> Apologies!
>
> The object was not a stone. It was human.
>
> The error was excusable and quite understandable. There was a gentleman, a sightseer (bald-headed), who happened to stand in a gap in the line of bluestones at the instant of the click-click of the passing plane. His shadow was like that of a stone; his head top looked like polished dolerite. "Vertical object, height 5 ft 10 ins", recorded the machine.

Mark Brader

[More Aviation Hearsay?]

<mnetor!spectrix!clewis@seismo.CSS.GOV>
Wed Oct 8 12:04:57 1986

I understand and appreciate your comments in the mod.risks about nth party/ hearsay stuff. But, from the examples you gave, in case you are really looking for some aviation accidents partially due to obedience to the "book", here are two - both commercial accidents at Toronto International (Now Pearson International). Both from MOT (then DOT) accident investigations:

About 15 years ago a Air Canada DC-8 was coming in for a landing. At 60 feet above the runway, the pilot asked the co-pilot to "arm" the spoilers. The co-pilot goofed and fired them. The aircraft dropped abruptly onto the runway, pulling about 4 G's on impact. At which point one of the engine/pylon assembly tore away from the wing - this was an aircraft defect because the engines were supposed to withstand this impact - a 6 G impact is supposed to shear the mounting pins. Not aware of this fact, the pilot performed what the book told him to do - go around for another try. He only made it halfway around - the pylon had tore away a portion of the fuel tank and the aircraft caught fire and crashed in a farmer's field killing all aboard.

In retrospect, the pilot should have stayed on the ground, contrary to the book. Many would have survived the fire on the ground. However, it was difficult to see how the flight crew could have realized that the aircraft was damaged as it was in the short time that they had to decide. The spoiler arming system was altered to make this more unlikely.

The second incident was about 8 years ago - on a Air Canada DC-9 taking off. During take off one of the tires blew throwing rubber fragments through one of the engines. One of these fragments damaged a temperature sensor in the engine, causing an "engine fire" indication to come on in the cockpit. The pilot did what the book said, "abort takeoff", even though he was beyond the safe stopping point. The aircraft slid off the end of the runway and into the now infamous 50 foot deep gully between the runway and the 401 highway. The fuselage broke in 2 places, causing one death and several broken bones and minor back injuries.

In retrospect, if the pilot had not aborted takeoff, he would have been able to take off successfully and come around for reasonably safe landing, saving the aircraft and preventing any injuries. However, there was absolutely no way that they could have determined that the engine was not on fire.

Results:
- in spite of the findings, I seem to remember that the pilot was suspended for some time.
- Recommendations:

    - filling in the gully - not done
  - cutting grooves in the runways for improved braking - not done yet,
    but the media is still pushing the MOT. (I'm neutral on this one,
    the MOT has some good reasons for not doing it)
  - cleaning the tarmac of burned rubber - only done once if I recall
    correctly.

As a counter example, I offer you another:

It had become common practise for twin-otter pilots to place the props in full reverse pitch while landing, instants before actually touching down. This had the effect of shortening the landing run considerably over the already short runs (twin-otter is STOL). However, due to a number of accidents being traced to pilots doing this too soon - eg: 50 feet up, the aircraft manufacturer then modified the aircraft so as to prevent reverse pitch unless the aircraft was actually on the ground.

(The above, however, is from a newspaper, and would bear closer research).

Search RISKS using swish-e

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 78

Thursday, 9 October 1986

Contents

- On models, methods, and results
  Bob Estell
- Fault tolerance vs. verification experiments
  Nancy Leveson
- The second Tomahawk failure
  PGNeumann
- Re: Overrides and tradeoffs
  Eugene Miya
  Herb Lin
- Software getting old
  Ady Wiernik
- Rebuttal -- Software CAN Wear Out!
  George Cole
- "Obsolescence" and "wearing out" as software terms
  Dave Benson
- Obsolesence and maintenance - interesting non-software anecdote
  Jon Jacky
- FAA - Plans to replace unused computers with new ones
  McCullough
- Info on RISKS (comp.risks)

On models, methods, and results

"ESTELL ROBERT G" <estell@nwc-143b.ARPA>
7 Oct 86 08:14:00 PST

Models, methods, and conclusions are, as Dave Benson pointed out, connected. Let's explore that more.

There are at least two kinds of models: formal [well understood mathematics], and informal [heuristics or things that just work];
and there are two approaches to model usage: induction, and deduction;
finally, a model can be in the mind of the user implicitly, or explicitly.

I'll conjecture that there is a high positive correlation between "informal, induction, and implicitly", and an even higher positive correlation between "formal, deduction, and explicitly." Though a good logician would wince at the notion of "deduction from heurustics" I'll further conjecture that all eight cells in this model are populated, albeit unevenly.
I'm not arguing that this SHOULD be so; I'm NOT saying HOW computer models should work; I'm just admitting how lots of people do think. In many cases "fuzzy" [or worse] is a good description; e.g., we "prove" things based on a few examples, or the opinion of an authority, often in the absence of good scientific theory, or verifiable facts.
Humans satisfice a lot; even those of us who are primarily analytical are also a bit pragmatic; we rarely "undo" something that works just because we don't understand it completely. ["Undo" means "retract" and forfeit the benefits. Often we do "dissect" after the fact, for better understanding.]

I think that some of the back-and-forth in RISKS is between groups at ends of a spectrum; at one end, there are those who are using informal models, based on experience, who induce conclusions from them implicitly; at the other end, are those who yearn for formal models, with results deduced explicitly. My sympathies lie with both groups; I too yearn to understand; but IF forced to choose, I'd take results now, and wait for understanding. But in fact, I believe there's a middle course; I believe we are already achieving some successes; and I believe we have some under-standing. I personally have experienced much bettter results using high order languages, and writing modular code; hence my "understanding" that these techniques are "better" than some alternatives. If this experience is not universal, that's no surprise either. The most intricate piece of code I ever wrote was an I/O driver that ran in diagnostic mode; it was very short, more or less modular, and in a mixture of assembly and machine language. It solved a problem so poorly understood that I got more credit for the project than I probably deserved.

As others in RISKS have pointed out, we need to take some care with words; else, we'll lose the ability to understand each other. OKAY, it was a mistake to ever think that anyone thought that "SDI software had to be perfect." I think that agreement represents enormous progress. Thank you.

Now, can we proceed to define "acceptable." [Or other terms.]
Can we begin to use some numbers? Can we remember that Hamming was right: "The purpose of computing is insight, not numbers." But can we have some numbers anyhow, just to help us understand?

Another analogy may help. In baseball, a pitcher is credited with a "perfect game" if no opposing batter reaches first base safely. He doesn't have to strike out all 27 batters; or retire each with only one pitch, by getting them to hit easy pop-ups, or easy grounders. [NOTE that real purists could argue endlessly about these two cases; which is better? striking out all 27, which requires at least 81 pitches? or retiring all 27 with only 27 pitches?]
If the definition of "perfect" is arbitrary, it doesn't matter too much, since there are so few perfect games. Wins, strike-outs, earned run average, and other metrics usually help us decide who the great pitchers are.

One case we've been discussing [Viking Lander] seems to indicate that the software was "successful" while admitting that it had flaws. Without some metrics, we'll rehash our differing opinions endlessly.

One closing thought about models. It's a fact that induction is always at risk of being upset by "the next case." It's also true that deduction is not able to prove anything outside the scope of the axiom set on which it is based. At their extremes, the one is fragile; the other, sterile.
Life should be both robust and fertile; it's more fun that way.
A judicious blending of the analytical and the practical can give us some clues to how near the extremes we're operating.

Bob

Fault tolerance vs. verification experiments

<leveson@sei.cmu.edu>
Thu, 9 Oct 86 11:06:38 edt

In Brian Randell's message (RISKS 3.77) he says:

>Indeed, it strikes me that it would be very interesting if the planned
>experiments that Nancy refers to were to cover various verification and
>testing, as well as, fault tolerance experiments.

We are indeed doing this in the latest of our series of experiments on software fault tolerance (the first of which was reported in TSE in January and the second, which involves fault detection, is currently being written up -- both were done jointly with John Knight at the University of Virginia). The experiment in question, which is being conducted by Tim Shimeall (a Ph.D. student of mine at UCI) includes comparison of software fault tolerance methods with various expensive validation methods including sophisticated testing and code reading by hierarchical abstraction.

We would like to include formal verification also, but have not found funding and other support for this yet. John McHugh at RTI may join in the experiment by providing versions of the program using IBM Clean Room techniques (a form of formal verification along with software reliability growth modeling is used in the development of the programs), but again we have not yet found funding.

The programs involve a battle management application (the Viking problem did not turn out to be appropriate) which is based on a real program developed by TRW (who are partially funding the experiment). Twenty versions of the program are currently under development, and we should be able to report some results by next summer.

Nancy Leveson
Info. & Computer Science Dept.
University of California, Irvine

The second Tomahawk failure

Peter G. Neumann <Neumann@CSL.SRI.COM>
Thu 9 Oct 86 15:25:40-PDT

Apparently the second Tomahawk test failure was due to a bit dropped by the hardware, resulting in the accidental triggering of the ABORT sequence. (Readers may recall that a parachute opened and the missile landed safely.)

Re: Overrides and tradeoffs (Jerry Leichter)

Eugene Miya <eugene@AMES-NAS.ARPA>
7 Oct 1986 1247-PDT (Tuesday)

I would just like to point out the software engineering dilemma of dealing with RISKy systems.

> Subject: Overrides and tradeoffs
> Accidents in general are fairly low-probability events.

This is in inverse proportion to the use of effort taught by the 90-10 rule (90% of the time is used by 10% of the code and other variants). RISKs are a case where the remaining 10% taking the other 90% of the time. Perhaps, we should think about the 10% first (error handling) and worry about the high probability events last? I don't know, but I did give an example earlier where this was the case.

--eugene miya, NASA Ames Research Center
{hplabs,hao,dual,ihnp4,decwrl,allegra,tektronix,menlo70}!ames!aurora!eugene

---

Overrides and tradeoffs (Risks 3.74)

<LIN@XX.LCS.MIT.EDU>
Thu, 9 Oct 1986 09:06 EDT

The best work I have seen on this stuff is work by Kahneman and Tversky, who identify two "heuristics" that people use to estimate probability -- availability and representativeness. Availability is the ease with which one can remember a particular event, so that if you have direct experience with something, it is more salient in your mind, and thus you think that it is more likely. Representativeness is using the extent to which the features of a particular situation match your general conception of a class of situations to determine the probability that the situation is a member of the class, rather than using other kinds of information to make that judgment.

A great review book on the subject is "Judgment under Uncertainty: Heuristics and Biases", edited by Daniel Kahneman, Paul Slovic, Amos Tversky, Cambridge University Press, 1982.

Availability and representativeness explain A LOT!

[NOTE: By the way, speaking of REPRESENTATIVEness, Herb has accepted a full-time position for one year as a "Congressional Fellow", sponsored by the American Association for the Advancement of Science. (The purpose of the Congressional Fellowships is to take professional scientists, engineers, social scientists, etc. and expose them to the policy-making process and in turn contribute some scientific expertise to the decision-making process.) It seems to me wonderful that he is willing to spend a year in such an enterprise. We expect Herb to continue participating in RISKS as an integral part of his job, and hope to have some inside RISKS SCOOPS. Perhaps RISKS can even have an impact on Congress! PGN]

Software getting old

Ady Wiernik <ady%taurus.BITNET@WISCVM.WISC.EDU>
Sun, 5 Oct 86 16:02:07 -0300

It has recently been suggested in this forum that software whose environment changed over time (requiring a change in the functional specification) might become "old" and "rotten". One example given was that of financial software which can't handle high inflation rate (having insufficient number of digits in various total fields).

Well, here in Israel we have already gone through two currency changes: in 1977 we changed the currency from Lira to Shekel (which was 10 Liras) and in 1985 we change it again from Shekel to new Shekel (which equals 1000 old Shekels). These changes affected every piece of financial software in the market, and before each change there was usually a period in which financial software had to be adjusted to have more digits in total fields. In addition to this, we had gone from an inflation rate whose peak was 21% per month to an average 2% per month inflation.

Most packages survived the changeovers rather easily.

The morale of this is - even if the environment changes drasticly, software doesn't have to die. It all depends on how much you are willing to pay the physicians (maintenance programmers). Only the software which was bad to start with (i.e. didn't sell well) will die due to natural selection.

Ady Wiernik Tel-Aviv Univ.

Rebuttal -- Software CAN Wear Out!

<Cole.pa@Xerox.COM>
9 Oct 86 10:05 PDT

Software as it exists in the programming language (and mathematical statements) is theoretically perfect -- a Platonian Ideal. Yet there is a risk that the software as it is embedded in the hardware will become distorted and "worn out". Between background radiation, hardware failures causing bit-changes (the resistor lets too little or too much current through, causing a "1" to be read as a "0"), and people-caused hardware failures (bent pins, crimped cables, etc.), there is the chance for distortions in the software. In "Bad Bits" (Scientific American, Feb. 1980, p. 70, there is a reference to radiation failures (presumably from background radiation) causing random failures -- I believe the figures are 3,000 / million hours of operation in a 256k charge-coupled device.

I do not think these sources are currently a major part of the "software failures" in the industry; design, specification and maintenance problems seem to be far more prevalent because of the lack of attention paid to human engineering problems -- the basic presumption seems to be that Murphy's Law doesn't hold for people (programmers or administrators or scientists). As computing machines become more "dense" though, this real possibility of unpredictable failure ought to be considered.

George S. Cole, Esq.
GCole@sushi.stanford.edu

"Obsolescence" and "wearing out" as software terms

Dave Benson <benson%wsu.csnet@CSNET-RELAY.ARPA>
Mon, 6 Oct 86 18:16:53 pdt

JS>Dave Benson nicely identifies a distinction between becoming obsolete
JS>and wearing out, and argues that only the former applies to software.

Thank you, but not quite. Software can also become lost, dissipate or in various other manners disappear. One way this might occur is if all copies of the media containing the bit patterns wore out. But in most situations, software becomes obsolete before it disappears.

JS>There is another effect that isn't exactly captured by the words "to
JS>become obsolete." ...
JS>Some people have in mind the impairment and diminished usability
JS>caused by this effect when they use words like "wears out" or "rots".

Software suffers so-called "maintenance" since the changing requirements require modification. This is common enough in other engineered artifacts: Coal-fired steam plants have exhaust scrubbers added, etc. The diminished usability in software is caused by the rapidly changing external conditions, thus "obsolesence" is an entirely appropriate term. The fact that the re-engineering of the artifact in the attempt to keep the artifact current is poorly done only causes the artifact to become obsolete more rapidly than if the re-engineering was done well.

JS>I guess we need a plain English word for it so that neophytes won't
JS>think that computers that haven't been oiled properly rub too hard
JS>on the bits.
How about "obsolete"? Here are some examples. Some are fact, others

fiction, still others opinion. Decide whether the word fits before
coining a new term.

Arpanet is rapidly showing its obsolescence under the dramatically increased traffic. While the obsolescence of the Enroute Air Traffic Control System is appearent to the controllers, it is judged that providing computers with 3 times current speed will keep the system operational until the year 2010. The financial transaction system of the Bank of Calichusetts is showing its obsolesence by the large losses to so-called computer criminals. Unix will be obsolete by the year 2010, then being replaced by Yaos, which is currently in advanced engineering at the Yet Another Company. The LGP-30 is an obsolete computer. Sage is an obsolete software system. SDI sofware will be obsolete before it is written.

I shudder at the thought that this may become so popular that the gerund "obsolescing" will appear on RISKS.

Obsolesence and maintenance - interesting non-software anecdote

Jon Jacky <jon@june.cs.washington.edu>
Tue, 7 Oct 86 22:49:43 PDT

Hammersmith Hospital, in London England, closed down its research cyclotron last year. The cyclotron was the first ever to be dedicated to medical research and applications (mostly, production of radioactive tracer chemicals and treatment of cancer with neutron beams), and began running in 1955. According to one of the physicists on the staff, who gave a seminar at the University of Washington yesterday, an important factor in the decision to close the facility was that the original designer is scheduled to retire this year, and he is the one person who really understands how to keep it going and modify it. England's Medical Research Council (or MRC, sort of like NIH in this country) is building a replacement cyclotron at a different site at the cost of many millions of pounds.

-Jonathan Jacky
University of Washington

[There is of course an analagous problem in software. PGN]

FAA - Plans to replace unused computers with new ones

<mccullough.pa@Xerox.COM>
Tue, 7 Oct 86 11:32:07 PDT

Federal officials say a problem-plagued air traffic control system installed at many U.S. airports four years ago probably will be replaced before most of the equipment is ever used. The multimillion-dollar system was supposed to make radar screens clearer, help track aircraft that do not carry radar signal equipment and otherwise relieve some of the load on the existing system. But engineers have been stumped by programming problems that have rendered the

Sensor Receiver and Processor System, or SRAPS, virtually useless, the Orange County Register reported Saturday. The agency expects a new $500 million Westinghouse system ordered 2 1/2 years ago to arrive in November or December for testing, said FAA engineer Marty Pozesky. Known as the ASR-9, for Airport Surveillance Radar, the system will affect virtually every U.S. airport, he said, adding it may be four years before it is fully operating. The SRAPS computers were purchased in 1981 from the now-defunct Sperry Univac Information Storage Systems. Researchers at the successor company, Sperry, have continued to seek a solution to the software problems, but no longer have the help of FAA engineers, Pozesky said. "We don't think the solution will be there, so we have really stopped searching," Pozesky said by telephone Saturday from his Silver Spring, Md., home.

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 79

Sunday, 12 October 1986

Contents

- China Air incident... the real story
  Peter G. Trei
- Air-Traffic Control Spoof
  Peter G. Neumann
- Aviation Accidents and Following Procedures (RISKS-3.77)
  Matthew Waugh
- DC-9 crash again
  Peter Ladkin
- Info on RISKS (comp.risks)

China Air incident... the real story

Peter G. Trei <OC.TREI@CU20B.COLUMBIA.EDU>
Mon 13 Oct 86 01:04:22-EDT

Excerpted from 'Tumbledown Jumbo', an article in the Oct 86' issue of FLYING magazine, concerning the China Airlines 006 incident of Feb 86.

ellipses and contractions in [square brackets] are mine.

...

At one point the autothrottle brought the engines back to about zero thrust. ...as the throttles came forward again, the number-four engine did not respond. The flight engineer ... told the captain that the engine had flamed out.

Maximum restart altitude is 30,000 feet [the plane started at 41,000]. The captain told the first officer to request a lower altitude. He then told the engineer to attempt a relight, even though the plane ... was still at 41,000. The restart attempt was unsuccessful.

The captain ... released the speed and altitude hold on the autopilot. The autopilot was now programmed to maintain pitch attitude and ground track. The airplane continued to lose speed gradually ... and the captain eventually disengaged the autopilot completely and pushed the nose down.

At the same moment, the airplane yawed and rolled to the right. The captain's attitude indicator appeared to tumble [as did two backups].

The airplane had now entered the clouds. At the same time ... the other three engines quit.

[paragraph omitted, describing speed varying between Mach .92 and 80 knots, as crew attempts recovery under up to 5G accelerations.]

After ... more than two minutes, the 747 emerged from the clouds at 11,000 feet and the captain was able to level it by outside reference. Coincidentally, he felt that the attitude indicators 'came back in' at this point. [engines 1,2, & 3 restart themselves, and 4 responds to a checklist restart].

Initially the captain decided to continue ... [but it was noticed that] the landing gear was down and one hydraulic system had lost all its fluid. ... the captain decided to land at San Francicso. The plane operated normally during descent, approach and landing.

[Later analysis showed that engine four had NOT flamed out, but just stuck at low thrust due to a worn part. The others were also responding to the throttles very slowly, a common problem at 41,000 feet. The NTSB inquiry concluded that...] the captain had become so preoccupied with the dwindling airspeed that he failed to note that the autopilot, which relied on ailerons only, not the rudder, to maintain heading, was using the maximum left control-wheel deflection available to it to overcome the thrust asymmetry due to the hung outboard engine. When the right wing nevertheless began to drop, ... the captain didn't notice the bank on the attitude indicator ... . When he did notice it, he refused to believe what he saw. At this point, ... the upset had begun and the captain and first officer were both spatially disorientated.

[...]

Once the erroneous diagnosis of a flameout had been announced, ... the captain placed excessive reliance on the autopilot.... When he finally disengaged it, and put himself 'back into the feedback loop' it was at a critical moment, and he could not adjust quickly enough to the unexpected combination of control feel and instrument indications to prevent the upset.

END OF QUOTATIONS.

The rest of the article is devoted to RISKS-style analysis of use of automatic systems. To give a more down-to-earth (pun intended) analogy, suppose your car was equipped with an AI 'drivers assistant', which handled all normal highway driving. Suppose further, at night, with you drowsy and at 60 mph, the right front wheel blows out. The AI blasts the horn to alert you, and applies substantial left torque to the steering wheel to keep it straight. You realize your in trouble, grab the wheel, and turn off the AI. The wheel immediatally jumps out of your hands to the right (you didn't know how much torque the AI was applying), and the car swerves off the road...

The use of automated systems to handle routine operations of critical systems, with dangerous situations suddenly dumped in the hands of human operators, presents a new Risk... that they may not fully understand the ramifications of the problem during the critical transition time.

A co-worker of mine who has worked in both the Navy and civilian nuclear programs tells me that Navy reactor systems are designed to keep humans in the loop. The only thing the automated systems can do without a person is 'scram' or shut down the reactor. Changes in power level, opening and shutting valves, pulling control rods, operating pumps, etc, must be performed by the people constantly tending the reactor. Thus, the system cant very easily spring surprises on the operators.

Air-Traffic Control Spoof

Peter G. Neumann <Neumann@CSL.SRI.COM>
Sat 11 Oct 86 20:03:57-PDT

Some of you may have missed a recent set of rather serious breaches of the integrity of the air-traffic control system. It is another important instance of a masquerading spoof attack typified by the Captain Midnight case (although via voice rather than digital signals). [Again note the October 86 issue of Mother Jones noting similar vulnerabilities and the ease of performing attacks.]

Washington Post, 8 October 1986

MIAMI -- A radio operator with a ``bizarre sense of humor'' is posing as an air traffic controller and transmitting potentially dangerous flight instructions to airliners, and pilots have been warned about it, an Federal Aviation Administration spokesman said. Two fake transmissions have occurred in the last week, and one caused a premature descent, said Jack Barker of the FAA's southern region in Atlanta. ``There have been no dangerous incidents, but the potential for danger is there. It's more an annoyance than a safety problem,'' Barker said from an FAA meeting in Washington. Barker said the operator uses two frequencies that air traffic controllers use to tell pilots how to approach Miami International Airport. The transmissions began Sept. 25, and the last was Friday [3 Oct], he said.

Aviation Accidents and Following Procedures (RISKS-3.77)

<ihnp4!houxm!mtuxo!pegasus!phoenix!poseidon!popeye!naples!mjw@ucbvax.Berkeley.EDU>
Fri, 10 Oct 86 11:50:44 PDT

The accident report involving a British Airways 737 at Manchester Airport was released recently. The aircraft suffered an engine compressor failure on take-off. The aircraft instruments indicated something else (I'm a little hazy about exactly what, I think it was a tire burst), and standard operating procedure was to turn clear of the runway, basically I believe to clear the runway for other traffic. This the pilots did, bringing the wind, which had been dead ahead to blow from the now burning engine and wing, onto the fuselage. Multiple lives were lost, etc.

It would appear from this that had the pilots performed an abort and maintained the runway, all that would be required for safety reasons, the deaths could have been reduced or avoided. However the operating procedure, for operational (not safety) reasons mandated otherwise and worsened an otherwise pretty terrible situation.

UUCP : {ihnp4|mtuxo}!naples!mjw Matthew Waugh
ATTMAIL: attmail!mjw AT&T IS, Lincroft, N.J.
Telephone : (201) 576-3362

DC-9 crash again

Peter Ladkin <ladkin@kestrel.ARPA>
Fri, 10 Oct 86 14:50:49 pdt

Danny Cohen's point about accuracy is well taken. The incident I was trying to refer to was the crash of Eastern 212, a DC-9, in Charlotte, N.C. I apologise to Risks readers for not confirming this before posting.

Danny and I have exchanged letters on the issue of *deliberate override*. Danny considers the action of turning off the LAAS to be both non-deliberate and not an override. I still consider it both deliberate and an override. It seems to hinge on whether habitual actions can be described as deliberate, and on whether not following prescribed procedure upon receipt of a warning can be considered an override.

Peter Ladkin

Search RISKS using swish-e

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 80

Wednesday, 15 October 1986

Contents

- US Navy reactors
  Henry Spencer
- Data Protection Act Risks
  Lindsay F. Marshall
- Is Bours(e)in on the Menu?
  Martin Minow
- Re: Software Wears Out
  anonymous
- Info on RISKS (comp.risks)

US Navy reactors

Henry Spencer <decvax!utzoo!henry@ucbvax.Berkeley.EDU>
Tue, 14 Oct 86 17:56:36 edt

> A co-worker of mine who has worked in both the Navy and civilian
> nuclear programs tells me that Navy reactor systems are designed to keep
> humans in the loop. The only thing the automated systems can do without
> a person is 'scram' or shut down the reactor... Thus, the
> system can't very easily spring surprises on the operators.

A probable contributing factor here is that the US Navy's submarine people do not trust automation at all in crucial roles. For example, US subs have no autopilots, even though they spend most of their time at constant speed and depth. They are "flown" manually at all times. This is not so much a matter of keeping the operators alert and informed as it is a matter of complete distrust of complexity and automation in submarines. This is a significant constraint on submarine design, in fact. Modern subs generally have a fairly symmetrical set of vertical and horizontal fins at the tail. Looked at from behind, it's a cross shape. There would be advantages to using an X shape instead, just shifting the whole cluster 45 degrees: this would permit grounding the sub on the bottom without damage to the bottom fin, and would permit docking against a straight dock without worries about banging one of the horizontal fins against the dock. The US Navy does not think highly of the idea, because it would require a mixing box of some kind (which could be purely mechanical!) to turn the horizontal and vertical control inputs into rudder/elevator motion. That's how deep the distrust of complexity runs. I'm not surprised that they have manually- controlled reactors.

The USN also has an outstanding reactor safety record -- no big accidents, no serious radiation releases -- with a stable of reactors comparable in numbers (although not in output) to the entire US nuclear-power industry. They are very fussy about materials, assembly, and operator training.

Henry Spencer @ U of Toronto Zoology
{allegra,ihnp4,decvax,pyramid}!utzoo!henry

[Intriguing. I have frequently heard it said -- by Nancy Leveson and others -- that the nuclear power technology is so sensitive that they feel they cannot afford to use computers! PGN]

Data Protection Act Risks

"Lindsay F. Marshall" <lindsay%cheviot.newcastle.ac.uk@Cs.Ucl.AC.UK>
Wed, 15 Oct 86 14:36:27 gmt

Police find a Catch 22 for data victim - From The Guardian

The police are ready to challenge the new right to compensation guaranteed by the Data Protection Act to people injured through the passing of inaccurate information. Hertfordshire Police, which wrongly suggested to Tayside Regional Council that a woman it was considering appointing had a criminal record, has denied that the woman has any claim to compensation. Under the Data Protection Act, all agencies - including the police - which hold information electronically are liable to damage claims for any harm which inaccuracies create for people on their records. But Hertfordshire Police has produced a Catch 22 defence. In a letter to the woman's solicitor, the force suggests that the woman has no claim to compensation. The police now conceded that the woman does not have a criminal record but go on to argue that she is therefore not on their records. As she is not a "data subject" she cannot be eligible for compensation.

Mr. Eric Howe, the Data Registrar, said yesterday that he would resist such an interpretation of the act. One problem for the woman, Mrs Anne Trotter, of Kirriemuir, Tayside, would be the cost of the court action. There is no legal aid in such cases. The Data Registrar can initiate criminal prosecutions but cannot sponsor civil actions. The case would cost over 1,000 pounds.

The mistake happened earlier this year. Tayside Regional Council social work department, which was considering appointing Mrs. Trotter to a special fostering programme for delinquent teenagers, followed the recommended procedure of checking the criminal records of its applicants. The authority wrote to the police in Hertfordshire, where Mrs Trotter had lived for a period, and was informed that two separate sets of "convictions are recorded against Anne Trotter, who appears identical with the applicant." They involved thefts in Newcastle upon Tyne in 1942 and theft and false pretences in Newcastle in 1947.

Anne Trotter's maiden name was Lawson until she married in 1954. In 1942 she was 15 years old and was still at school in Arbroath. The police were given her maiden name. Mrs Trotter was so upset by the incident that she decided to drop her application and take up a temporary teaching post. She asked the social services department for a copy of the police letter and, unusually, was given one. The right of access to such letters does not come into force until November next year.

Later, after hearing about the Data Protection Act, she took it to a solicitor in Dundee. He wrote to the Hertfordshire Police on July 3 asking for compensation. The police replied on July 8, denying responsibility. The force said its letter had only said the Newcastle offender "appears identical with the applicant." The letter went on to claim: "The fact of the matter is that your client is not a data subject within the terms of the Data Protection Act as it is now clear ... that no records are held in respect of your client."

Mr Kevin Veal, the solicitor, sent a second letter which said: "It seems to use that insufficient care was given to the issue. For example, it must have been obvious to anyone compiling the report that a young girl born in 1927 under the name of Lawson could not have been convicted under the name of Trotter in 1942.

The case is made more complicated by the fact that the police supplied the information on April 21 but the compensation provisions of the act only came into force on May 11. There was no retraction, however, until July 8 and no attempt by the police in the letters to use the May 11 date as the reason for not providing compensation.

Is Bours(e)in on the Menu?

<minow%regent.DEC@decwrl.DEC.COM>
15-Oct-1986 1530

(Martin Minow, DECtalk Engineering ML3-1/U47 223-9922)

BEAR MARKET MEANS BARGAIN FOR DINERS
By Paul Lewis

(reprinted without permission from the New York Times News Service)

PARIS - The two hungry diners sat down, turned expectantly to a flickering computer screen on a nearby stand and began studying the latest quotations. The news seemed ominous. Making money would not be easy in today's luncheon market.

The scene was La Connivence, a small new bistro-style restaurant at 6 Rue Feydeau, a stone's throw from the Paris Bourse, or stock exchange. As with stocks on the exchange, the laws of supply and demand determine the price diners at La Connivence pay for a meal. (The name, La Connivence, means complicity, with the slightly shady overtones appropriate for a gambling den of sorts.)

As patrons place their orders in the austere ground-floor dining room, one of the owners, Jean-Claude Trastour, enters them into a computer which promptly adjusts the menu prices to reflect demand. Popular dishes, like popular stocks, go up in price while less popular ones decline.

Timorous diners may choose to pay the quoted price for a dish at the moment they order it. That is called eating on the march comptant, or cash market. If the price rises while these diners are tucking in, they have done very well for themselves. If the price falls, they get indigestion. It is the safe way to eat - safe and dull.

More adventurous folks play the futures market, the march a terme, agreeing to pay the price quoted when they call for the check at the end of their meal. Naturally, they hope the price will have fallen by that fateful moment. But hopes may be dashed by a flurry of buying, and the price may easily shoot up. Worse indigestion.

The newly seated diners began preparing their gambling strategy by reading the trends. They saw that the prices of several dishes had already fallen by close to 6 francs--the limit for price changes up or down in any one eating-trading session. (A dollar is worth about 7 francs.) That left little room for further decline. There would be no point in ordering any of those dishes, no matter how delectable--unless, of course, the diner was more interested in eating than in successful speculation.

The computer screen flashed chute du filet mignon, indicating that the price of that choice steak had already fallen 5 francs, to 50 francs a serving. A veal casserole with herbs had slipped 4 francs, to 48 francs. A rack of lamb chops for two, down 10 francs, was priced to sell for 110 francs a serving. As for the haddock, the computer reported a "sharp fall" of 5 francs a portion, to 57 francs.

Other dishes were doing better. The screen showed that a "stampede" of orders for lotte had pushed the price of that pleasant Mediterranean fish up 4 francs to 62 francs a portion, making it an interesting speculation. If diners played the forward market, the price might be substantially lower when the time came to pay; of course, it could still rise another 2 francs before reaching the 6 francs ceiling.

Occasionally, a diner's greed is outweighed by the thought of what he would have to eat to turn a profit. An example: "Victorious advance of the stuffed pigs' trotter," the computer flashed, marking it up 5 francs, to 43 francs. Surely it could only fall. But a lunch of pigs' feet?

In the end, the diners chose a conservative strategy, ordering the special of the day, saddle of lamb, on the marche a terme. The lamb was trading at 39 francs a portion; up a modest 2 francs for the day thus far.

The check arrived for the conservative diners: 228 francs for two, which is pretty good by Paris standards since it included a bottle of Beaujolais, a cheese-filled ravioli from the French Alps for a starter, homemade apple tart, and coffee. But the roast saddle of lamb stood at 38 francs, only a meager 1 franc cheaper than when it was ordered. Down the street, the Bourse was having one of its best days ever.

[Inside tip: Sell-SHORT-Ribs, Buy-LONGustine. Bon appetit! Pierre]

Re: Software Wears Out

Anonymous <[...]>
Mon, 13 Oct 86 08:15:06 [...]

[I have been rejecting almost all messages on this subject, in that (1) the topic was not converging, and (2) the discussion might better belong in SOFT-ENG@MIT-XX. But this somewhat historical note seems worth including -- along with this note explaining that I have been throttling other contributions. PGN]

I have to remain anonymous because my management lives in fear that someone who works for them may post something dumb. Herewith, I justify their most morbid fears.

The comments on software "wearing out" vs. becoming obsolete seem to me to be dancing around the issue. L.A. Belady and M.M. Lehman addressed this matter in a seminal paper: "Programming System Dynamics, or the Meta-dynamics of Systems in Maintenance and Growth" (IBM Research, RC 3546, Sept 17, 1971).

The authors maintain that systems do have a "lifetime," and so in that sense, they may be supposed to wear out, although they do not use that term; nor do they say that software becomes obsolete. Instead, their measure is entropy. When the programming system's entropy is low, its ability to do "work" on its environment is high, and vice-versa.

A system at release, or shortly thereafter, possesses low entropy. Maintenance and enhancement over time increase the entropy until the marginal cost of the next required set of fixes and/or enhancements approaches, say, the amounts expended on the system up to that point. Entropy is then high, and the system may be said to be "worn out."

This is at best a poor precis of a very elegant paper; the gentle reader is referred to the original for a deeper insight into the reasons why software wears out.

[Among all the complaints that software is static and -- in never changing -- should not be said to "wear out", we note that it is often NOT static, which is of course a large part of the problem. In the other hand one might say that the INTERFACE wears out rather than the software. But let us not quibble on this one any more. PGN]

Search RISKS using swish-e

Report problems with the web pages to the maintainer

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 81

Sunday, 19 October 1986

Contents

- System effectiveness is NOT a constant!
  anonymous
- Aircraft self-awareness
  Scott Preece
- Re: US Navy reactors
  Brint Cooper
  Eugene Miya
  Stephen C Woods
- Editorial on SDI
  Michael L. Scott
- Info on RISKS (comp.risks)

System effectiveness is NOT a constant!

<[anonymous]>
16 Oct 86 20:03:00 [...]

There seems to be a tendency in the current SDI debate to fall into an old engineering fallacy: that systems scale up linearly. Everyone seems to avoid this trap when talking about cost and effort--it seems to be well accepted that a 10-million line program is much harder than 10 1-million line programs--but (most) people are *not* avoiding the trap when they speak of SDI's effectiveness. A recurrent argument seems to be that "SDI will be 80% [to use a number currently being bandied about] effective against a Soviet attack of N missiles; thus the Soviets would have to build and launch 5N missiles in order to have N missiles reach their targets, which would be economically ruinous." The implicit assumption is that if SDI is x% effective against N, it will continue to be x% effective against N'. This is fallacious unless x is very close to 0 or 100%. Assuming 80% effectiveness and 1000 missiles, SDI stops 800. Using the reasoning above, against 2000 missiles, SDI would stop 1600; but this cannot be so. If 1000 missiles strains the system to the point that it can only stop 800, why would anyone think it could stop more when the number of missiles and decoys is doubled, straining the system's ability to identify, track, and destroy missiles at least twice as much? Or to put it another way, if SDI could stop 1600 out of 2000, shouldn't it be able to stop 1600 out of, say, 1800 (1800 is surely an easier problem than 2000!). Or turn the argument around: if SDI can stop 800 out of 1000--80% effectiveness--does this mean it can stop only 80 out of a 100-missile attack? Or 8 out of a 10-missile attack?

When anyone says that SDI will have such-and-such effectiveness, they must be made to state the assumptions used to calculate that effectiveness. Otherwise the numbers are meaningless.

Aircraft self-awareness

"Scott E. Preece" <preece%ccvaxa@GSWD-VMS.ARPA>
Tue, 14 Oct 86 10:15:09 cdt

A lot of recent RISKS messages have discussed one kind or another of aircraft accident. Many of the reports have included things like "The pilot thought [X] but in fact [Y]" or "[X] occurred, though the indications were that [Y] had occurred" or "[X], though there was no way for the flight crew to know that".

So, what's going on in the area of improving flight crew/control system awareness of the state of basic external structures? Is anyone considering whether the FAA should require external cameras or periscopes so that (for instance) the pilot could find out that her entire vertical stabilizer had fallen off or her starboard outboard engine exploded?

While there are many cases where the pilot would not, in any case, have time to check, there are also cases like the Japan Airlines crash where the plane stayed up for some time but the pilot had no way to determine the gross condition of the control surfaces. Some reports have said that that plane might have been saved if the pilot had known what he had to compensate for.

Given that we are depending more and more on automated controls, should we be spending more effort on sensors that can determine more basic kinds of information? Should the control surfaces be instrumented so that the flight controls can tell the captain "Oh, the starboard outboard engine is no longer on its pylon and the outer flaps on that wing seem to be missing." as opposed to current systems just recognizing the effects of that loss and trying to compensate, with the risk that the operator will be unaware of the magnitude of that compensation and forced to guess at the state of the aircraft by observing what the control system is doing to deal with the effects of that state ("Oh, I'm having to turn the rudder vigorously to port to maintain my heading; can't say why.").

scott preece, gould/csd - urbana
uucp: ihnp4!uiucdcs!ccvaxa!preece

Re: US Navy reactors

Brint Cooper <abc@BRL.ARPA>
Thu, 16 Oct 86 8:33:57 EDT

Henry Spencer writes:
> A probable contributing factor here is that the US Navy's submarine people
> do not trust automation at all in crucial roles... That's how deep the
> distrust of complexity runs. I'm not surprised that they have manually-
> controlled reactors.

Then, he observes:
> The USN also has an outstanding reactor safety record -- no big accidents,
> no serious radiation releases -- with a stable of reactors comparable in
> numbers (although not in output) to the entire US nuclear-power industry.
> They are very fussy about materials, assembly, and operator training.

Perhaps we should suspect that the safety record follows directly from
the suspicion?
Brint

RE: Reactors of the USN

Eugene Miya <eugene@AMES-NAS.ARPA>
Thu, 16 Oct 86 09:14:52 pdt

I generally concur with Henry Spencer's accessment. The USN is very conservative about its use of proven technologies and reliability (also notice all new Navy jets have two engines [exclude older A-4, A-7, and F-8s]). But, while the Navy's record is certainly outstanding, I must point out there is a question about "no big accidents."

One of the major contending theories on the loss of the USS Thresher in 1964 was sudden loss of reactor power. We will never really if this is the case, but it cannot ignored.

Excellent reading about the safety record, the conservativitism, and the development of the nuclear navy is found in the 700+ page unauthorized biography of Rickover.

--eugene

US Navy reactors [RISKS-3.80 DIGEST]

Stephen C Woods <scw@LOCUS.UCLA.EDU>
Fri, 17 Oct 86 11:43:17 PDT

There is another factor to consider here, redundancy. Submariners are ALL cross trained EXTENSIVELY (the ideal is that everyone can do everything, usually they come fairly close to the ideal).

Why, you may ask, does the Navy go to such lengths? The answer is fairly simple; these are WARSHIPS, they need to be able to function even after suffering SEVERE damage and heavy casualties. Just for normal day to day operations there are at least 2 people for every job (watch on and watch off), usually there are 3, often there are 4 or more.

The following from net.aviation may be of interest to you. (ESP the quote). You may be interested in the whole discussion there. [scw]

>From: wanttaja@ssc-vax.UUCP (Ronald J Wanttaja)
>Newsgroups: net.aviation
>Subject: Re: Problems with flying by the book (a pithy comment)
>Date: 14 Oct 86 15:58:15 GMT
>Organization: Boeing Aerospace Co., Seattle, WA

<> I understand and appreciate your comments in the mod.risks about nth party/
<> hearsay stuff. But, from the examples you gave, in case you are really
<> looking for some aviation accidents partially due to obedience to the
<> "book", here are two - both commercial accidents at Toronto International
<> (Now Pearson International). Both from MOT (then DOT) accident
<> investigations:
>
[...]

>"Rule books are paper: They will not cushion a sudden meeting of stone and
> metal."
> - Earnest K. Gann

Editorial on SDI

Michael L. Scott <scott@rochester.arpa>
Sat, 18 Oct 86 17:51:36 edt

The following is an op-ed piece that I wrote for the Rochester, NY, DEMOCRAT AND CHRONICLE. It appeared on page 4A on September 29, 1986.

'STAR WARS' CAN'T SUCCEED AS SHIELD, HAS OFFENSIVE CAPABILITY

Can the Strategic Defense Initiative succeed? The answer depends critically on what you mean by success. Unfortunately, the public perception of the purpose of SDI differs dramatically from the actual goals of the program.

In his original "Star Wars" speech, President Reagan called upon the scientific community to make nuclear weapons "impotent and obsolete." He has maintained ever since that this is the SDI goal: to develop an impenetrable defensive shield that would protect the American population from attack. With such a shield in place, nuclear missiles would be useless, and both the United States and the Soviet Union could disarm.

Can such a shield be built? The most qualified minds in the country say "no." In an unprecedented move, over 6,500 scientists and engineers at the nation's research Universities have signed a statement indicating that "Anti-ballistic missile defense of sufficient reliability to defend the population of the United States against a Soviet attack is not technically feasible." The signatures were drawn from over 110 campuses in 41 states, and include 15 Nobel Laureates in Physics and Chemistry, and 57% of the combined faculties of the top 20 Physics departments in the country. Given the usual political apathy of scientists and engineers, these numbers are absolutely staggering.

The obstacles to population defense include a vast array of problems in physics, optics, astronautics, computer science, economics, and logistics. Some of these problems can be solved with adequate funding for research; others cannot. Consider the single subject of software for "Star Wars" computers. As a researcher in parallel and distributed computing, I am in a position to speak on this subject with considerable confidence. The computer programs for population defense would span thousands of computers all over the planet and in space. They would constitute the single largest software system ever written. There is absolutely no way we could ever be sure that the software would work correctly.

Why not? To begin with, we cannot anticipate every possible scenario in a Soviet attack. Human commanders cope with unexpected situations by drawing on their experience, their common sense, and their knack for military tactics. Computers have no such abilities. They can only deal with situations they were programmed in advance to expect. Before we can even start to write the programs for "Star Wars," we must predict every situation that might arise and every trick the Soviets might pull. Would you bet the future of the United States that the Russians won't think of ANYTHING we haven't thought of first?

Even if we could specify exactly what we want the computers to do, the task of translating that specification into flawless computer programs would be beyond our capabilities for many, many years, possibly forever. Current and projected techniques for testing and quality control may reduce the number of flaws in large computer systems, but actual use under real-life conditions will always uncover further "bugs." (For details on the software problem, see Dr. David Parnas's article in the October 1985 issue of AMERICAN SCIENTIST.) The only way to gain real confidence in "Star Wars" software would be to try it out in full-scale nuclear combat. Such testing is clearly not an option.

But if effective population defense is impossible, why are we spending billions of dollars on SDI, and why are the Russians so upset about it? The answer is remarkably simple: because population defense is not the goal of SDI. The kinetic and directed energy devices being developed for the "Star Wars" program will have a tremendous range of uses in offensive weapons and in increasing the survivability of U.S. land-based missiles. The Soviets fear "Star Wars" for its first-strike capabilities. To make nuclear weapons impotent and obsolete, SDI would have to be perfect. To shoot down Soviet satellites, to thin out a pre-emptive strike on U.S. missile fields, or to develop exotic new weapons for the conventional battlefield, SDI will only need to succeed on a much more modest level.

By focusing public attention on the unattainable goal of population defense, the Administration has managed to avoid discussion of the more practical, immediate consequences of SDI research. The weapons developed for "Star Wars" will have a profound impact on both our war-fighting strategy and our treaty obligations. That impact should be the subject of public and Congressional debate. By pretending to develop a defensive shield, the President has fooled the American people into funding a program that is far less clear-cut and benign. In effect, he has sold a system we cannot build in order to build a system he cannot sell.

BYLINE:

Michael L. Scott is an Assistant Professor of Computer Science at the University of Rochester. His article was co-signed by 10 other faculty members [almost the entire department] and 36 doctoral students and researchers. The views expressed should not be regarded as the official position of the University of Rochester or of its Computer Science Department.

[We haven't had any RISKS mention of this topic in a long time. Perhaps it is time to dust it off again in the light of Reykjavik. The nature of the offensive capability is not a new issue, but is clearly an enormous potential RISK -- at least in the eyes of the Soviets. However, subsequent discussion on that issue probably belongs on ARMS-D. Let's once again try to stick to issues relevant to computers and related technologies. PGN]

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 82

Monday, 20 October 1986

Contents

- NASDAQ computer crashes
  - Jerry Leichter
  - Vint Cerf
- Sensors on aircraft
  - Art Evans
  - Henry Spencer
- Loss of the USS Thresher
  - John Allred
- Re: US Navy reactors
  - Henry Spencer
- Risks from Expert Articles
  - Andy Freeman
- Info on RISKS (comp.risks)

NASDAQ computer crashes

<LEICHTER-JERRY@YALE.ARPA>
20 OCT 1986 11:09:46 EST

OTC stock market - Computer problems snag trading

Computer problems halted trading for about three hours throughout the day Thursday [16 October 1986] in over-the-counter stocks listed through the National Association of Securities Dealers Automatic Quotation system. Craig Thompson, manager of marketing information for the National Association of Securities Dealers, said the system was shut down from about 11:05 a.m. to 2 p.m. EDT, then five minutes before the 4 p.m. closing due to a breakdown of equipment at its computer operations center in Trumbull, Conn. The exact nature of the problem had not been determined, Thompson said. "We don't think it will effect tomorrow's business as we hope it will be corrected by then," Thompson said.

{AP News Wire, 16-Oct-86, 16:48}

NASDAQ computer crashes

<CERF@A.ISI.EDU>
20 Oct 1986 06:47-EDT

Since so much of Wall Street operation is heavily dependent on automation and communication, it would be very interesting to know more about the causes and nature of the failure and how dealers/users coped with the outage. Obviously, neither Wall Street nor the economy collapsed, but it might be instructive to know whether the ability to accommodate the failure was a function of the length of the outage (how close to disaster did we actually approach? How much longer an outage could have been sustained without permanent damage?).

Vint Cerf

Sensors on aircraft

"Art Evans" <Evans@TL-20B.ARPA>
Mon 20 Oct 86 13:11:56-EDT

It's all well and good to propose a sensor that reports, "the left engine isn't there," or, "the left ailerons are gone," or whatever. But, how is the sensor to work? That is, just what do you propose to sense? Sure, you and I can look at the left wing and decide immediately, but what is the sensor to do? Moreover, how do you propose checking the reliability of a sensor that, in the nature of things, almost never does anything? I think these are hard problems.

As for the JAL 747 disaster -- the flight crew knew precisely what the problem was: With the loss of all three (or was it four?) hydraulic systems, they had no control whatsoever over any control services. They may not have known what caused the problem, but they were all too aware of the effects.

Aviation Week published the transcript of the cockpit voice recorder not too long after the accident, and it is the most terrifying such transcript I've ever read. The flight crew were dead, and they knew it. They were still flying around, but they were in effect test pilots in a new kind of aircraft no one had ever thought much about before. Their problem was simple: control pitch attitude (nose up or down) with power, and control direction with differential power (more power on one side than the other). Well, maybe with plenty of time to experiment someone might learn to fly a 747 that way. They tried, as long as they could, but they just weren't able to hack it. Most power adjustments produced oscillations in attitude that they were unable to damp out. Finally, it got away from them in a way they couldn't recover from, and they went down. A brave attempt at the probably impossible.

Art Evans

Aircraft self-awareness (Sensors on aircraft)

Henry Spencer <decvax!utzoo!henry@ucbvax.Berkeley.EDU>
Mon, 20 Oct 86 22:00:32 edt

I believe some of the DC-10 engineers proposed during development that it should have a set of video cameras viewing things like the wings and tail, so that the flight crew could get a look at the situation if they really needed to. (This is not as good as having it automatically brought to their attention, but many classes of problems would come to their attention quickly anyway...) The proposal was rejected, I believe on grounds of cost and weight.

In fairness, the only DC-10 crash I remember offhand where this might have helped was the Chicago engine-separation one, and it's not clear that the crew had time to study the problem. I don't know what the proposal had in the way of monitors, but for sheer reasons of panel space I suspect it would have been a switchable monitor rather than a bank of screens showing all views continuously. That crash happened fast; I doubt that information not available at a glance would have helped.

Henry Spencer @ U of Toronto Zoology
{allegra,ihnp4,decvax,pyramid}!utzoo!henry

Loss of the USS Thresher

John Allred <jallred@labs-b.bbn.com>
Mon, 20 Oct 86 13:31:40 EDT

Thresher, according to the information I received while serving on submarines, was lost due to a catastrophic failure of a main sea water valve and/or pipe, causing the flooding of a major compartment. The cause of the sinking was reported by the mother ship during the boat's sea trials. Scorpion, on the other hand, had no observer present. No reason of loss has been given to the public.

The loss of reactor power, in and of itself, should not have caused the loss of the Thresher. Boats are usually trimmed to be neutrally bouyant, so the loss of motiviation should not be fatal.

Does anyone out in netland have access to the report of the Thresher's loss? It would be good to hear the true story.

Re: US Navy reactors

Henry Spencer <decvax!utzoo!henry@ucbvax.Berkeley.EDU>
Mon, 20 Oct 86 22:00:42 edt

Brint Cooper suggests that the USN's excellent reactor safety record might stem from their deep distrust of automatic equipment. Personally, I think the connection is indirect. It's not at all obvious that manually-run reactors are safer than partly-automated ones. Humans are better at coping with unforeseen situations, *if* they truly understand the equipment they are controlling. If they're just being used as organic servomechanisms, then they are less reliable than automatic equipment, which does not get tired or bored (when things are going well) or frightened or tense (when they aren't). I suspect the USN reactor technicians have a pretty good understanding of their hardware, given the general atmosphere of great care surrounding USN reactors. However, servomechanisms are probably still safer when the problems have, in fact, been foreseen accurately. This is likely to be the case for the majority of problems.

The indirect connection I see is the obvious one: distrust breeds caution. Whether or not manually-operated reactors are safer than semiautomated ones, *any* equipment clearly is going to be safer when elaborate care is taken in materials, assembly, testing, crew training, and maintenance. A high-quality reactor run by carefully-trained humans is clearly safer than a slipshod one run by rusty machinery.

Eugene Miya notes that there is some doubt about the reactor being blameless in the loss of the Thresher. True; I should have noted that.

Steve Woods notes:

> There is another factor to consider here, redundancy [cross-training] ...
> ... these are WARSHIPS, they need to be able to function even
> after suffering SEVERE damage and heavy casualties...

While I tend to agree that cross-training is a good idea, it's actually not clear that the USN has thought this one through, for submarines in particular. It's not obvious to me that there is any likelihood of severe damage and heavy casualties in a nuclear sub without catastrophic hull damage as well. Nuclear subs generally do not have internal pressure bulkheads, as I recall, because there isn't enough buoyancy reserve for the sub to survive with a flooded section anyway. This means that a serious hull breach is quickly fatal.

Henry Spencer @ U of Toronto Zoology
{allegra,ihnp4,decvax,pyramid}!utzoo!henry

Risks from Expert Articles

Andy Freeman <ANDY@Sushi.Stanford.EDU>
Mon 20 Oct 86 11:45:32-PDT

Scott@rochester.arpa (Michael L. Scott) wrote the following in RISKS-3.81:

> Why not? To begin with, we cannot anticipate every possible scenario in a Soviet attack. Human commanders cope with unexpected situations by drawing on their experience, their common sense, and their knack for military tactics. Computers have no such abilities. They can only deal with situations they were programmed in advance to expect.

Dr. Scott obviously doesn't write very interesting programs. :-)

Operating systems, compilers, editors, mailers, etc. all receive input that their designers/authors didn't know about exactly. Some people believe that computer reasoning is inherently less powerful than human reasoning, but it hasn't been proven yet.

Most op-ed pieces written by experts (on any subject, supporting any position) simplify things so far that they're actually incorrect. The public may be ignorant, but they aren't stupid. Don't lie to them. (This is one of the risks of experts.)

It can be argued that SDI isn't understood well enough for humans to make the correct decisions (assuming super-speed people), let alone for them to be programmed. That's a different argument, and Dr. Scott is (presumably) unqualified to give an expert opinion. His expertise does apply to the "can SDI decision be programmed correctly?" question, which he spends just one paragraph on.

-andy

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 83

Tuesday, 21 October 1986

Contents

- Risks from Expert Articles
  - David Parnas
  - Herb Lin
  - Andy Freeman
- Loss of Nuclear Submarine Scorpion
  - Donald W. Coley
- Staffing Nuclear Submarines
  - Martin Minow
- An SDI Debate from the Past
  - Ken Dymond
- System effectiveness is non-linear
  - Dave Benson
- Stealth vs Air Traffic Control
  - Schuster via Herb Lin
- Missing engines & volcano alarms
  - Martin Ewing
- Info on RISKS (comp.risks)

Re: Risks from Expert Articles (RISKS-3.82)

<parnas%qucis.BITNET@WISCVM.WISC.EDU>
Tue, 21 Oct 86 09:39:51 EDT

Andy Freeman criticizes the following by Michael L. Scott, "Computers have no such abilities. They can only deal with situations they were programmed in advance to expect." He writes, "Dr. Scott obviously doesn't write very interesting programs. :-) Operating systems, compilers, editors, mailers, etc. all receive input that their designers/authors didn't know about exactly. "

Scott's statement is not refuted by Freeman's. Scott said that the computer had to have been programmed, in advance, to deal with a situation. Freeman said that sometimes the programmer did not expect what happened. Scott made a statement about the computer. Freeman's statement was about the programmer. Except for the anthropomorphic terms in which it is couched, Scott's statement is obviously correct.

It appears to me that Freeman considers a program interesting only if we don't know what the program is supposed to do or what it does. My engineering education taught me that the first job of an engineer is to find out what problem he is supposed to solve. Then he must design a system whose limits are well understood. In Freeman's terminology, it is the job of the software engineer to rid the world of interesting programs.

Reliable compilers, editors, etc., (of which there are few) are all designed on the basis of a definition of the class of inputs that they are to process. We cannot identify the actual indvidual inputs, but we must be able to define the class of possible inputs if we are to talk about trustworthiness or reliability. In fact, to talk about reliability we need to know, not just the set of possible inputs, but the statistical distribution of those inputs.

Dave Parnas

Risks from Expert Articles

<LIN@XX.LCS.MIT.EDU>
Tue, 21 Oct 1986 09:16 EDT

From: Andy Freeman <ANDY at Sushi.Stanford.EDU>

Operating systems, compilers, editors, mailers, etc. all receive input that their designers/authors didn't know about exactly.

When was the last time you used a mailer, operating system, compiler, etc.. that you trusted to work *exactly* as documented on all kinds of input? (If you have, pls share it with the rest of us!)

It can be argued that SDI isn't understood well enough for humans to make the correct decisions (assuming super-speed people), let alone for them to be programmed. That's a different argument, and Dr. Scott is (presumably) unqualified to give an expert opinion. His expertise does apply to the "can SDI decision be programmed correctly?" question, which he spends just one paragraph on.

You are essentially assuming away the essence of the problem by asserting that the specs for the programs involved are not part of the programming problem. You can certainly SAY that, but that's too narrow a definition in my view.

Re: Risks from Expert Articles

Andy Freeman <ANDY@Sushi.Stanford.EDU>
Tue 21 Oct 86 14:40:48-PDT

Herb Lin writes:

> When was the last time you used a mailer, operating system, compiler, etc.. that you trusted to work *exactly* as documented on all kinds of input? (If you have, pls share it with the rest of us!)

The programs I use profit me, that is, their benefits to me exceed their costs. The latter includes their failures (as well as mine). A similar metric applies to weapons in general, including SDI. (Machine guns jam too, but I'd rather have one than a sword in most battle conditions. The latter are, for the most obsolete, but there aren't perfect defenses against them.)

Lin continued with:

> You are essentially assuming away the essence of the problem by asserting that the specs for the programs involved are not part of the programming problem. You can certainly SAY that, but that's too narrow a definition in my view.

Sorry, I was unclear. Specification and implementation are related, but they aren't the same. There are specs that can't be implemented acceptably (as opposed to perfectly). Some specs can't be implemented acceptably in some technologies, but can in others. (This can be context dependent.) Dr. Scott's expertise applies to the question of whether a given spec can be programmed acceptably, not whether there is an spec that can be implemented acceptably. Much of the spec, including the interesting parts of the definition of "acceptable", is outside CS, and (presumably) Dr. Scott's expertise.

Another danger (apart from simplification to incorrectness) of expert opinion articles is unwarranted claims of expertise. Dr. Scott (presumably) has no expertise in directed energy weapons yet he claims that they can be used against cities and missiles in silos. Both proponents and opponents of SDI usually agree that it doesn't deal with cruise missiles. If you can kill missiles in silos and attack cities, cruise missiles are easy.

-andy

Loss of Nuclear Submarine Scorpion

Donald W. Coley <coley@SCRC-VALLECITO.ARPA>
Tue, 21 Oct 86 12:38 EDT

This is in response to John Allred's comments about the loss of both the Thresher and the Scorpion (RISKS-3.82).

> Date: Mon, 20 Oct 86 13:31:40 EDT
> From: John Allred <jallred@labs-b.bbn.com>
> Subject: Loss of the USS Thresher

> Thresher, according to the information I received while serving on submarines, was lost due to a catastrophic failure of a main sea water valve and/or pipe, causing the flooding of a major compartment. The cause of the sinking was reported by the mother ship during the boat's sea trials.

Just to confirm what John stated, fracture of a hull-penetration fitting, at the weld between the flange and the pipe, quickly flooded the engineering spaces. The sinking had nothing to do with the reactor.

> Scorpion, on the other hand, had no observer present. No reason of loss has been given to the public.

Scorpion was in very high speed transit, westbound in one of the submarine transit lanes, when she struck a previously uncharted undersea mountain. The speed of the collision was "in excess of forty miles per hour" (probably closer to sixty). It was the very high speed that had rendered her (acoustically) blind; unable to see the obstacle in her path. True, no observer was present, but a lot of people did get to hear the result. The "days spent searching for the lost sub" were just to avoid revealing how accurate our tracking capabilities were. All the Navy brass knew within the hour, exactly what had happened and exactly where.

Staffing Nuclear Submarines

Martin Minow, DECtalk Engineering ML3-1/U47 223-9922 <minow%regent.DEC@decwrl.DEC.COM>
21-Oct-1986 1457

Disclaimer: a few months ago, my knuckles were rapped when I incorrectly cited a study on airline safety. Please be warned that I know absolutely nothing about nuclear submarines and am using the ongoing discussion about automatic controls for nuclear reactors (on submarines) only as a starting place for a wider discussion.

From the discussion on Risks it seems that, while automatic controls may do a satisfactory job of running the reactor in normal circumstances, people will still be needed to run the reactor when the automatic controls malfunction.

Adding automatic controls adds weight (and probably noise), making the ship less effective.

Adding automatic controls to a nuclear submarine's reactor frees personnel for other tasks. But, there isn't much else for them to do (they can hardly chip rust on the deck), so they'll get bored and lose their "combat readiness."

Relying on totally manual control keeps the crew alert and aware of the action of the reactor. It also keeps them busy.

In other words -- and I think this is directly relevant to Risks -- there are times when external factors make it unwise to automate a task, even when it can easily be done.

Martin

An SDI Debate from the Past

"DYMOND, KEN" <dymond@nbs-vms.ARPA>
21 Oct 86 11:03:00 EDT

While looking something up in Martin Shooman's book on software engineering yesterday, I came across the following footnote (p.495):

> Alan Kaplan, the editor of Modern Data magazine, posed the question, "Is the ABM system capable of being practically implemented or is it beyond our current state-of-the-art ?" The replies to this question were printed in the January and April 1970 issues of the magazine. John S. Foster, director of the Office of Defense Research and Engineering, led the proponents, and Daniel D. McCracken, chairman of Computer Professionals against ABM, led the opposition.

It's startling that the very question that so interests us today was put 15 or so years ago; to make it the exact question, all you have to do is change the 3 letters of the acronym. And this was 3 (?) generations ago in computer hardware terms (LSI, VLSI, VHSIC ?) and some indeterminate time in terms of software engineering (I can't think of anything so clear-cut as circuit size to mark progress in software). International politics, however, seems not to have changed much at all.

I'll try to track down those articles (Modern Data no longer exists having become Mini-Micro Systems in 1976), but in the meantime can anyone shed light on this debate from the dim past ?

(BTW, Shooman comments "Technical and political considerations were finally separated, and diplomatic success caused an abrupt termination of the project." p. 498)

System effectiveness is non-linear

Dave Benson <benson%wsu.csnet@CSNET-RELAY.ARPA>
Mon, 20 Oct 86 16:01:06 pdt

I agree with Anon that overall system effectiveness is non-linear:

>If 1000 missiles strains the system to the point that it can only
>stop 800, why would anyone think it could stop more when the number of
>missiles and decoys is doubled, straining the system's ability to
>identify, track, and destroy missiles at least twice as much?

The more reasonable (and conservative) assumption is that the SDI system would stop ZERO missles when faced with, say, 2000 targets. Case in point is revision n of the US Navy Aegis system -- seems that being designed to track a maximum of (17) targets, when there are (18) targets the computer software crashed.

Any engineered artifact has design limits. When stressed beyond those limits, it fails. We understand this for civil engineering artifacts, such as bridges. Clearly this is not well understood for software engineering artifacts.

<Schuster.Pasa at Xerox.COM>
Tuesday, 21 October 1986 10:39-EDT

After reading the recent ARMS-D on the Stealth subject, particularly the interesting message from Bryan Fugate where he says that "stealth fighters and bombers have already gone into production", and in light of some of the recent aircraft collisions, I couldn't help but wonder if anyone has adequately considered the air traffic control consequences of not being able to get a radar fix on a large, rapidly moving aircraft in a high density air traffic area?

For that matter, what about ground-radar-assisted-landing in poor visibility at a military base?

Sometimes you want an aircraft to present a GOOD radar target. As I was writing this I thought of the answer, I guess. The stealth aircraft would have to have a strong beacon turned on in these circumstances. I guess it's easy to recreate a good target this way. All I can say is that the beacon had better be working in the circumstances I described.

Missing engines & volcano alarms

Martin Ewing <mse%Phobos.Caltech.Edu@DEImos.Caltech.Edu>
Tue, 21 Oct 86 13:41:58 PDT

We visited New Zealand a few years ago and went to the major skiing area on the North Island (the name escapes me). It is built on the slopes of an active volcano. There were prominent warnings for skiers of what to do in case of an eruption alarm. (Head for a nearby ridge. Don't try to outrun the likely mud/ash slide coming down the hill.)

How do they get the alarm? There is an instrument hut at the lip of the crater connected to park headquarters by a cable. The instruments measure some parameter(s) or other. (heat, acceleration, pressure, ?) When something crosses a threshold, the warning alarms on the ski slopes are set off automatically.

In fact, someone admitted, what would probably happen is that the explosion would destroy the hut and cut the cable. Loss of signal is probably as good a diagnostic as anything else.

I can imagine a display on the DC-10 instrument panel inscribed with the outline of the aircraft. Little red lights come on when you lose continuity on a wire to an engine, aileron, etc. - like what happens when you leave your door open on a Honda Civic. What you do with this data is another matter.

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 84

Tuesday, 22 October 1986

Contents

- Risks of using an automatic dialer
  Bill Keefe
- Re: Missing engines & volcano alarms
  Eugene Miya
- False premise ==> untrustworthy conclusions
  Martin Harriman
- USN Automated Reactors
  Dan C Duval
- Keep It Simple as applied to commercial nuclear power generation
  Martin Harriman
- Works as Documented
  Martin Minow
- Re: Editorial on SDI
  Michael L. Scott
- Risks from Expert Articles
  Herb Lin
- Stealth vs. ATC / SDI Impossibility? / Missing Engines ?
  Douglas Humphrey
- Info on RISKS (comp.risks)

Risks of using an automatic dialer

Bill Keefe <keefe%milrat.DEC@decwrl.DEC.COM>
Wednesday, 22 Oct 1986 10:09:16-PDT

I wonder if it's significant that they are willing to talk about payment for aggravation but not for lost business. Unfortunately, it was not reported whether the failure was due to a hardware or software problem.

Computerized Sales Call Gets Stuck, Ties Up Phone for Three Days

GREENWICH, Conn. (AP) - A shipping broker who does all his work on the phone says he lost at least one deal because a computerized sales pitch called him nearly every two minutes for 72 hours, tying

up his lines.
The voice-activated computer message bedeviling Joern Repenning was shut off Monday after he had complained to New York Telephone's annoyance bureau, the Better Business Bureau, AT&T, police and the state attorney general.
The problem was in a computer at Integrated Resources Equity Corp. in Stamford, said William Banks, an employee of the company. The repeated calls blocked all other incoming calls to Repenning's office with a busy signal.
``There was no way we could conduct business,'' Repenning said. ``We can't shut off our telephone. That's our business.''
He said he lost at least one deal because he could not reply by a certain deadline on a shipping-cargo transaction.
Integrated is willing to talk with Repenning about payment for aggravation he suffered, Banks said.

Re: Missing engines & volcano alarms

Eugene Miya <eugene@AMES-NAS.ARPA>
Wed, 22 Oct 86 09:34:51 pdt

Martin Ewing gives an example of "absence of signal" as an indication that something maybe wrong. He concludes by precisely indicating the problem but glossing over with "What you do with this data is another matter." This last statement is unacceptable completion of the argument for aircraft manufacturers.

This is precisely the problem with planes, spacecraft, and other highly constained systems. How do we adequately know something, almost as bad: how do we know our instrument is not malfunctioning? Do we perenially "tap" the instrument? Designers of aircraft prefer "indicator/effector" systems, not to put just "indicators" into planes. "Great, my wings fell off" so what are you going to do?

There is a wind tunnel across the street from where I lunch. This tunnel has a set of sensor wires which enter a plate. This struck me as the nerve system of the wind tunnel when I first saw it. How inadequate this appears. The metal hull of the tunnel isn't a sensory tool like our skin (able to sense, heat, pressure, and other things to a much better precision). Some day perhaps.

On the posting on the safety of Stealth aircraft: I was visiting a friend on the day of the recent non-crash of the non-existent F-19. We were assured (not-assured?) by authorities [friend lived within a few miles of the non-site] that, since the non-F-19 only flew at night, it ALWAYS flew with a radar detectable chase plane (not a non-plane).

--eugene miya

False premise ==> untrustworthy conclusions

Martin Harriman <"SRUCAD::MARTIN%sc.intel.com"@CSNET-RELAY.ARPA>
Wed, 22 Oct 86 14:54 PDT

There seems to be a misconception floating around in RISKS regarding the degree of automation in Navy and civilian nuclear reactors. Civilian reactors are not significantly different than Navy reactors in this respect; both types of reactors have a single form of automated control. Both Navy (propulsion) and civilian (electric power generation) reactors have a reactor protection system--a system rather like a circuit breaker that automatically shuts the reactor down if some parameters (such as reaction level or temperature) exceed defined limits. If you've ever seen the reactor jargon "scram" or "trip" (as in, "we had three unplanned trips this year"), that's what is being referred to.

Everything else is manual, in either system.

At least in civilian systems, this system is tested regularly (planned trips), and the reactor's responses noted. I am not sure if the Navy has planned trips; I know they have unplanned trips often enough to annoy the reactor operators (the scram alarm is a *very* loud klaxon in a *very* small compartment).

Reactors are not a good paradigm for a debate on the risks of automated controls. Arguments based on the safety record of one class of reactors versus another will miss the point; the reactors differ in many interesting respects (training, discipline, nature of the task, ...), but the nature of the control mechanisms is not one of them.

USN Automated Reactors

Dan C Duval <dand@tekigm.TEK.CSNET>
21 Oct 86 12:10:27 PDT (Tue)

Arguments over whether the Navy's choice to NOT use automated safety systems on USN reactors are overlooking one major point, in that the choice of using or not using any safety equipment of any kind also has to meet a weight/benefit tradeoff.

If you design a reactor with built-in automated safety features, you have the weight of the reactor, the weight (and bulk) of the safety systems, the reactor operators (and the systems to support them, such as galleys, bunk space, food stores, etc), and the personnel to maintain the safety equipment (with support for them as well).

A "manual" reactor requires only the reactor and the operators (plus their support).

Adding the automated safety gear adds weight, requiring a larger boat, a larger power plant, more support for boat and crew, etc, all for no added war-fighting capability. Meantime, adding training to a human being does not add appreciable weight to the human being, nor require further support systems.

Though this weight consideration is paramount for subs, it holds as well
for surface ships (or "targets", as my ex-submariner buddy calls them.)
Thus, I think the argument that the USN doesn't trust automation is
weakened, since the USN also has other things to worry about than just
the automated safety vs non-automated safety tradeoff.

This weight-consideration argument also has some bearing on the aircraft
sensor question. More weight in sensors means a larger plane, more systems
that can break, more potential for overlooking problems during maintenance,
and more ways to confuse the flight crew. (Scenario: Crew cannot see wing
to tell if engine has fallen off, but sensor says it has; did it fall off
or did the sensor fail? Did anyone ever see the movie where the flight crew
shut down their last remaining engine because coffee, spilled into the control
panel, caused the "Engine Fire" warning to sound? So we have sensors to
check the sensors, to check those sensors, etc.)

Dan C Duval, Tektronix, Inc
uucp: tektronix!tekigm!dand

Keep It Simple as applied to commercial nuclear power generation

"Martin Harriman" <"SRUCAD::MARTIN"@sc.intel.com>
Fri, 17 Oct 86 17:05 PDT

I think it might be rather amusing if the nuclear power generating plants
in the US were all run by some (reasonably competent) admiral. Oh well...

The nuclear power (design) industry--the folks who design the nuclear
steam supply systems and their controls--uses a very similar approach to
that used in the Navy. The automated controls on the reactors I am
familiar with are limited to the reactor protective systems--the system(s)
that detect a fault condition, and trip the reactor. These systems are
kept very simple (on the same principle as keeping a circuit breaker as
simple as possible for the job it does).

Control of reaction rate and profile is accomplished through manual adjustments
of the control rods and the water chemistry.

The reliability of this system (and its safety) depends on the quality of the
reactor operator (that is, the power company operating the reactor). One of
the more encouraging signs in recent years has been the NRC's willingness to
suspend the operating licenses of operators who have poor safety records:
the TVA suspension is the most obvious.

--Martin Harriman, Intel Santa Cruz

Works as Documented

Martin Minow, DECtalk Engineering ML3-1/U47 223-9922 <minow%regent.DEC@decwrl.DEC.COM>

22-Oct-1986 0842

> When was the last time you used a mailer, operating system, compiler,
> etc.. that you trusted to work *exactly* as documented on all kinds of
> input? (If you have, pls share it with the rest of us!)

The problem is not that the software (etc.) works as documented, but whether it works as we *expect* it to.

This distinction has wider applicability. We *expect* SDI to protect us from a Russian missile attack. SDI is *documented* to protect some large percentage of our missiles from a Russian missile attack.

Martin.

Re: Editorial on SDI

<scott@rochester.arpa>
Wed, 22 Oct 86 11:51:50 edt

RISKS-3.82 contains a response from Andy Freeman to an editorial I posted to RISKS-3.81. Andy and I have also exchanged a fair amount of personal correspondence in the past couple of days. In that correspondence he maintains that I have disguised a political argument as expert opinion. This from his posting to RISKS:

> Most op-ed pieces written by experts (on any subject, supporting any
> position) simplify things so far that they're actually incorrect. The
> public may be ignorant, but they aren't stupid. Don't lie to them.
> (This is one of the risks of experts.)

I do not believe that I have oversimplified anything. I certainly haven't lied to anybody (let's not get personal here, ok?).

When technical arguments disagree with government policy, it is standard practice to dismiss those arguments as "purely political." Almost everything that a citizen says or does in a democratic society has political overtones, but those overtones do not in and of themselves diminish the technical validity of an argument. "The emperor has no clothes!" can be regarded as a highly political statement. It is also technically accurate.

In my original editorial, I declared that we could not be certain that the software developed for SDI would work correctly, 1) because we don't know what 'correctly' means, and 2) because even if we did, we wouldn't be able to capture that meaning in a computer program with absolute certainty. Andy takes issue with point 1). My words on the subject:

> Human commanders cope with unexpected situations by drawing on their
> experience, their common sense, and their knack for military
> tactics. Computers have no such abilities. They can only deal with
> situations they were programmed in advance to expect.

This is the statement Andy feels is 'actually incorrect'. His words:

> Operating systems, compilers, editors, mailers, etc. all receive input
> that their designers/authors didn't know about exactly. Some people
> believe that computer reasoning is inherently less powerful than human
> reasoning, but it hasn't been proven yet....
>
> It can be argued that SDI isn't understood well enough for humans to
> make the correct decisions (assuming super-speed people), let alone
> for them to be programmed. That's a different argument and Dr. Scott
> is (presumably) unqualified to give an expert opinion.

Very true, the designers of everyday programs don't know about their input *exactly*, but they *are* able to come up with complete characterizations of valid inputs. That is what counts. The "inputs" to SDI include virtually anything the Soviets can do on the planet or in outer space. It does not require an expert to realize that there is no way to characterize the set of all such actions. A command interpreter is free to respond "invalid input; try again"; SDI is not.

I stand by the technical content of my article: SDI cannot provide an impenetrable population defense. Impenetrability requires certainty, and that we can never provide. Though the White House has kept debate alive in the minds of the public, it is really not an issue among the technically literate. Almost no one with scientific credentials is wiling to maintain that SDI can defend the American population against nuclear weapons. There are individuals, of course (Edward Teller springs to mind), but in light of the evidence I must admit to a personal tendency to doubt their personal or scientific judgment. Certainly there is no groundswell of qualified support to match the incredible numbers of top-notch physicists, engineers, and computer scientists who have publically declared that population defense is a myth.

What we do see are large numbers of individuals who believe that the SDI program should continue for reasons *other* than perfect population defense. It is possible to make a very good case for developing directed energy and kinetic weapons to keep the U.S. up-to-date in military technology and to enhance our defensive capabilities.

My editorial is not anti-SDI; it is anti-falsity in advertising. Those who oppose SDI will oppose it however it is sold. Those who support it will find it very tempting to allow the "right" ends to be achieved (with incredible budgets) through deceptive means, but that is not how a democracy is supposed to work. Let the public know what SDI is all about, and let us debate it for what it is.

Risks from Expert Articles

<LIN@XX.LCS.MIT.EDU>
Tue, 21 Oct 1986 22:43 EDT

LIN@XX.LCS.MIT.EDU (Herb?) writes:

> When was the last time you used a mailer, operating system, compiler, etc.. that you trusted to work *exactly* as documented on all kinds of input? (If you have, pls share it with the rest of us!)

> From: Andy Freeman <ANDY at Sushi.Stanford.EDU>
> The programs I use profit me, that is, their benefits to me exceed their costs. The latter includes their failures (as well as mine). A similar metric applies to weapons in general, including SDI.

But you can bound the costs of using a faulty mailer. You can't with missile defense for population.

> Dr. Scott's expertise applies to the question of
> whether a given spec can be programmed acceptably, not whether there is an spec that can be implemented acceptably. Much of the spec, including the interesting parts of the definition of "acceptable", is outside CS, and (presumably) Dr. Scott's expertise.

Are you saying that computer scientists should not be calling attention to the problem of writing specifications? Or that they have no expertise in knowing the consequences of faulty specs? I think quite the contrary -- computer scientists know, probably better than anyone else, how important the specs are to a functional program. I agree that CS background does not grant people particular knowledge about which specs are proper, but in my view CS people are entirely proper to holler about lousy specs and what would happen if they were bad.

> Another danger (apart from simplification to incorrectness) of expert opinion articles is unwarranted claims of expertise. Dr. Scott (presumably) has no expertise in directed energy weapons yet he claims that they can be used against cities and missles in silos.

Reports that space-based lasers can be used against cities were recently published, and a fairly simple order of magnitude calculation that anyone can do with sophomore physics suggests that city attack with lasers is at least plausible. You're right about silos.

> Both proponents and opponents of SDI usually agree that it doesn't deal with cruise missles. If you can kill missles in silos and attack cities, cruise missles are easy.

Hardly. The problem with cruise missiles is finding the damn things. Cities and silos are EASY to find.

Stealth vs. ATC / SDI Impossibility? / Missing Engines ?

Douglas Humphrey <deh@eneevax.umd.edu>
Wed, 22 Oct 86 12:52:44 EDT

This is kind of a grab bag of responses to the last RISKS.

Stealth vs. ATC - The general public does not seem to know a lot about the

Air Traffic Control system and how it works. In controlled airspace such as around large airports, a Terminal Control Area (TCA) is defined into which only aircraft equipped with a Transponder may traverse. In reality, the rules and flavors concerned with this whole process are very complex and aren't needed here. If you are really interested, go to Ground School. The transponder replies to the interrogation of the ATC radar providing at least a bright radar image, and in more sophisticated systems the call sign of the aircraft, heading, altitude, etc. Thus, the concept of Stealth vs. ATC is not real. If the stealth aircraft is flying under Positive Control of ATC, then it will have the transponder. If it does not have one, then it better stay out of busy places or it is illegal and the pilot sure as hell will have his ticket pulled.

[Peter Ladkin also responded on this point. However, if the stealth plane is foreign/unfriendly/hostile/sabotage-minded/..., and NOT flying under postive control of ATC, then this argument does not hold. PGN]

SDI Impossibility? - I have a good background in physics, computing (software and vlsi hardware) and a lot of DEW (Directed Energy Weapons), and I have yet to hear ANYONE explain WHY SDI is impossible. I hear all this about the complexity of the software, but I used to be part of a group that supported a software system of over 20 million lines of code, and it rarely had problems. Admittedly, we wrote simulators for a lot of the load since we did not want to try experimental code out on the production machines, but we never had a simulator fail to correctly simulate the situation. There were over 100 programmers supporting this stuff, and it was properly managed and it all worked well. Is someone suggesting that the incoming target stream can not be simulated ? Why not ? We do it now on launch profile simulations involving the DEW (Distant Early Warning) network and a lot of other sensor systems. Is someone suggesting that PENAIDS (Penetration Aids) can not be simulated ? Why not ? We do it now also. Worst case studies just treat all of the PENAIDS as valid targets. If you can intercept THAT mess, then you can stop anything !

I get the feeling that people are assuming that the SDI software is going to be one long chunk of code running on one machine and that if it ever sees anything that is not what it expects its going to do a HALT and stop the entire process. Wrong. I wouldn't build a game that way, much less something like SDI ?

So. The Challenge. People out there who think it is Impossible, please identify what is impossible. Pointing systems ? Target acquisition ? Target Classification ? Target descrimination ? Destruction of the targets ? Nobody is saying that it is easy. Nobody is saying that our current level of technology is capable of doing it all perfectly. But it sure isn't (in my opinion) impossible.

[We've gone around on this one before. DEH's message is somewhat fatuous, but needs a serious response. Before responding further, make sure you have read the Parnas Papers from American Scientist, Sept-Oct 1985, also reprinted in ACM Software Engineering Notes October 1985, and the Communications of the ACM, December 1985. But remember that we never seem to converge in these discussions. Parnas does not PROVE that SDI is IMPOSSIBLE. He gives some good reasons to worry about the software. No one else can prove that it CAN BE IMPLEMENTED to satisfy rigorous requirements for reliability, safety, security, nonspoofability, etc., under all possible attack modes and environmental circumstances -- even with full-scale deployment in real combat. Especially when operating under stressed conditions, things often fail for perverse reasons not sufficiently anticipated. (That should be particularly clear to long-time readers of RISKS.) Think about OVERALL SYSTEM TESTING in the absence of live combat as one problem, among others. Remember, this Forum exists as part of a social process, and contributions according to the masthead guidelines are welcome. But SDI debates seem to degenerate repeatedly into what seems like religious wars. So bear with me if I try to close the Pandora's box that I have again reopened. I would like to see some intelligent open discussion relating to computers and related technologies in SDI, but perhaps that is a futile wish. But once again, much discussion has taken place before, on both RISKS and ARMS-D. New RISKS participants might want to check back issues. See the summary issues at the end of Volumes 1 and 2 noted above, and the end of Volume 3 -- which will happen soon. Computer relevance to RISKS, else to ARMS-D. PGN]

Missing Engines - In most aircraft the loss of a major component of the control system is pretty obvious, generally announced by an abrupt change in the flight characteristics of the aircraft. Same would go for the loss of an engine. I am not sure why a pilot would need a video monitor to tell him that Number 2 just fell off the wing, or that he no longer has a left horizontal stabilizer. He will no doubt understand this by the way the aircraft is acting. Most pilots have a good understanding of Why they are flying and How, and are able to discern the condition of their aircraft from how it behaves. Certainly I know of Airline pilots who have been able to tell by the handling of a DC-9 that a cargo door was partially open, even though the indicator in the cockpit said it was closed.

[See above note from Dan Duval.]

I might mention that the landing gear might be a good place for some sort of camera system. Pilots get rather paranoid about the state of the landing gear when they fail to get 3 green lights up in the cockpit.

Doug Humphrey
Digital Express Inc.

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 85

Thursday, 23 October 1986

Contents

- On the Risk of Discussing SDI
  Craig Milo Rogers
- SDI Impossibility
  Douglas Humphrey
- Swedish Vulnerability Board Report on Complex System Vulnerabilities
  Chuck Youman
- Re: Thresher
  David Feldman
- Stealth and ATC
  Dan Melson
- Inoperative components
  Peter Ladkin
- Info on RISKS (comp.risks)

On the Risk of Discussing SDI

Craig Milo Rogers <ROGERS@B.ISI.EDU>
23 Oct 1986 17:52:08 PDT

The moderator recently requested intelligent open discussion relating to computers and related technologies in SDI. I believe that there has instead been too much discussion of computers and SDI.

The hardware and software issues raised by Parnas and others are interesting. They are complex, they defy simple quantification, and they relate directly to the work of many of the readers of this digest.

Yet, there are much simpler and more easily discussed problems with SDI. SDI provides minimal protection to Europe. SDI does not appear to provide protection against nuclear weapons launched at the US from off-shore submarines. Bombs can be smuggled into the US via, say, Canada, and reassembled in the hearts of our cities. Clearly, if you heed these arguments, SDI in no way makes nuclear waepons "impotent and obsolete".

By focusing our attention and that of the general public on computer-related SDI arguments, we run the *risk* of diverting attention from more important issues. We as computer technologists are raising the (weak, esoteric) issues with which we are familiar, when we as intelligent, informed citizens should be raising more general questions (perhaps precisely because we *are* less familiar with them).

There is a risk in introducing computers into a discussion in which they are not really relevant. It is not enough to be able to discuss an issue intelligently. One must also know when it is intelligent to raise the issue in the first place. (By the way, it is not clear to me that this message qualifies, either).

Craig Milo Rogers

[This issue reaches a relative high mark for noninclusion of messages, as I have omitted several on this topic. However, this one gets accepted -- because it is sound, objective, and coherent, and does not violate the other requirements. I have stated before that it is impossible to draw a line around "computer relevance". Craig's point is well taken. By the way, I squelched the discussions between Michael Scott and Andy Freeman (plus a comment from Herb Lin) which were getting to third-order arguments and re-reinterpretations. (Both of the main participants still feel they have further clarifications to make.) However, I urge you all to take more care in your INITIAL statements. That can do wonders at staving off lengthy iterations. PGN]

SDI Impossibility

<LIN@XX.LCS.MIT.EDU>
Thu, 23 Oct 1986 08:47 EDT

From: Douglas Humphrey

Swedish Vulnerability Board Report on Complex System Vulnerabilities

Chuck Youman <m14817@mitre.ARPA>
Thu, 23 Oct 86 13:52:32 -0400

The October issue of Signal magazine contains an article by Thomas Osvald on "Computers, Vulnerability and Security in Sweden." It describes a number of projects carried out by the Swedish Vulnerability Board. Of particular interest to RISKS is a project that addressed the vulnerability problems associated with the complexity of EDP systems. Mr. Osvald writes:

> A system becomes too complex when nobody can intellectually
> understand and comprehend it. Thus, a company will not change a
> system because secondary effects cannot be foreseen. The board
> concluded that one of the problems of conventional, administrative,
> complex systems is that it is difficult or even impossible to
> change these systems in an orderly, controlled way. On the other
> hand, there is a rapid increase in the change rate in our society

> in general and a correspondingly increasing demand for flexibility
> in information systems.

> Therefore, it must be accepted that programs are for standard or
> nonrecurrent use with an ever shorter life expectancy. However,
> data that are the raw material of information will not change as
> quickly as the processing rules. Data are therefore the resource
> that has to be cultivated, protected, tended, preserved, and developed.
> This approach supports recent developments of systems design methods,
> such as fourth generation languages, data dictionaries, and data base
> techniques.

Unfortunately, the article does not include a bibliography. Does anyone out in RISKS-land know if a English translation of this report exists?

Charles Youman (youman@mitre.arpa)

Re: Thresher

David Feldman <feldman%dartmouth.edu@CSNET-RELAY.ARPA>
Wed, 22 Oct 86 02:34:25 edt

A friend of my dad's who served in the submarine service once told me his "version" of the events on the Thresher:

Water had gotten into a compartment (or at least onto a sensor) in the reactor unit, and that caused the reactor to scram. (According to him, this type of shutdown is unconditional and irreversible on USN subs). When the ballast tanks were blown, for some reason the delivery pressure of the air that cleared the ballast tanks came in higher than normal, and caused a greater temperature drop at the valves. The valves froze open, allowing all of the air to escape, leaving the Thresher defenseless.

Note: this is second hand from one submarine officer.
Dave Feldman
feldman@dartvax.edu

Stealth and ATC

Dan Melson <crash!pnet01!dm@nosc.ARPA>
Thu, 23 Oct 86 01:03:13 PDT

If it exists, they are hardly going to put it into heavily travelled airspace over high population areas, where everybody can see it.

As for radar signatures, civilian ATC relies upon a mode 3/a transponder, and targets are generated on our PVD's (primarily) as a result of that. If they want the aircraft visible to civil radar, they simply turn the transponder on.

(There are large areas of restricted airspace and MOA's (Military Operations Areas) where the military does it's own operations without hindering civil ATC, and if it exists, would guess that most stealth flights are within such areas)

The above information is non-classified, freely available to any private pilot.

DM

Inoperative components

Peter Ladkin <ladkin@kestrel.ARPA>
Thu, 23 Oct 86 18:28:22 pdt

Doug Humphrey wonders whether aircraft need cockpit warnings to tell of major failure modes. The answer seems to be yes. Multi-engine aircraft instructors will tell you that a common occurrence with simulated engine failures in multi-engine aircraft is for the student to feather the prop on the good engine. The NTSB notes that this happens for real, too.

Peter Ladkin
ladkin@kestrel.arpa

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 86

Sunday, 26 October 1986

Contents

- Addition to Census of Uncensored Sensors
  PGN
- Military vs. civilian automatic control systems
  Will Martin
- Re: System effectiveness is non-linear
  Scott E. Preece
- SDI assumptions
  Daniel M. Frank
- SDI impossibility
  David Chase
- Editorial on SDI
  Henry Spencer plus quote from David Parnas
- Info on RISKS (comp.risks)

Addition to Census of Uncensored Sensors

Peter G. Neumann <Neumann@CSL.SRI.COM>
Sun 26 Oct 86 15:37:02-PST

On 23 October 1986, a United Airlines Boeing 727 jet (UA 616, on a 20-minute flight from San Francisco Airport to San Jose) had the nose-gear indicator light stay on after takeoff, suggesting that the landing gear might not have retracted. The plane landed again at SFO at 7:48 AM (8 minutes after takeoff). The problem was later attributed to a malfunctioning nose gear indicator. [Source: San Francisco Chronicle, 24 October 1986, p. 30]

This is another example for the discussion on the risks of using sensors to detect aircraft behavior. Yes, if someone worries about this problem in advance, it is always possible to have redundant sensors and redundant indicators. (This is done in SRI's SIFT system [Software Implemented Fault Tolerance], a prototype flight-control system running at NASA Langley AFB.) The cost of that must be compared with the resulting costs. The total cost of even an 8-minute aborted flight (including fuel, landing fees, and delays -- with requeuing for takeoff) is nontrivial. There are of course all sorts of hidden costs in delays, such as the costs to passengers, and snowball effects if such a delay exhausts the pilot's flying time for the month and requires the location of another pilot! (That actually happened to me once...)

Military vs. civilian automatic control systems

Will Martin -- AMXAL-RI <wmartin@ALMSA-1.ARPA>
Fri, 24 Oct 86 15:11:22 CDT

Many good points have been brought out about the rationale for the Navy not having more automation in submarine control systems, including those on the nuclear reactors. I think that a particular aspect of this needs emphasis. There is a major difference in the basic concepts behind military systems vs. civilian implementations -- the mission may be more important than human life in the military environment, but never so in civilian situations. (Also the completion of the mission may be more important than the preservation of property or things in the military.)

It may well be necessary to "tie down the safety valve" to prevent a reactor scram on a submarine in order that the vessel complete the mission or action in progress, even if the inevitable result is death by radiation poisoning of the crew, or some fraction of them, or the destruction of the reactor or the vessel itself after it has completed the action it is required to perform. In a civilian situation, this is never true -- the production of electricity from reactor "X" can never be more important than the safety of the population around or even the operators of that reactor. (We ignore here the statistical probablity that the shutdown of reactor "X" will trigger a cascade of blackouts which will eventually result in some number of deaths due to related factors -- patients on the operating table, people trapped in elevators, etc.) In fact, the value of the reactor itself is more important than its continuing production of electricity -- it will be shut down to prevent faulty operation causing damage to itself. For a military device, completion of the wartime mission or task is often more important than the continued safety or preservation of the device itself.

In the light of this, it is reasonable to expect relatively elaborate, "idiot-proof", overriding automatic control systems in civilian installations, and the absence of such in military versions of similar devices (or perhaps the military system will have some for use only in peacetime or training situations, which can be switched off in wartime). It may be necesary to operate devices "outside their envelopes" or to violate various guidelines regarding safety in wartime missions. Also, of course, military systems should continue to be at least somewhat usable even after they have suffered damage and elaborate safety systems are merely something more that will be liable to damage in combat. It is not acceptable to have your power source turn off in the middle of a battle because a minor and easily-controlled fire burned nothing vital but only some part of an automatic safety system control circuit; it would be reasonable for a civilian reactor to shut down given the exact same situation.

Note please that I am not saying that wartime operation would routinely be done with a complete disregard for safety or that every mission is more important than the lives of the people carrying it out. But there will be certain exceptional circumstances where the missions are that important, where the sacrifice of some lives (and certainly some amount of property) is necessary for the achievement of larger goals. The military systems have to support both routine operation and these rare exceptions.

Will Martin

Re: System effectiveness is non-linear

"Scott E. Preece" <preece%mycroft@GSWD-VMS.ARPA>
Thu, 23 Oct 86 13:52:06 CDT

Dave Benson argues that it is more reasonable and conservative to assume that an overloaded system will fail entirely than to assume it will either perform at its design limit but no more or perform above its design limit.

That's unarguably the conservative assumption. I would deny that ANY assumption was reasonable, given only a performance ceiling and the knowledge that performance demand will exceed that ceiling. It is obvious that the system could be designed to perform in any of the suggested ways when unable to cope with load. Suggesting one response or another is simply expressing an opinion of the designers' competence rather than any realistic assessment of the risks of SDI. Given that neither the design nor the designers are determined yet, this is a silly exercise.

scott preece, gould/csd - urbana, uucp: ihnp4!uiucdcs!ccvaxa!preece
arpa: preece@gswd-vms

SDI assumptions

Daniel M. Frank <prairie!dan@rsch.wisc.edu>
25 Oct 86 20:35:15 GMT

It seems to me that much of the discussion of SDI possibilities and risks has gone on without stating the writers' assumptions about the control systems to be used in any deployed strategic defense system.

Is it presumed that SD will sit around waiting for trouble, detect it, fight the war, and then send the survivors an electronic mail message giving kill statistics and performance data? Much of the concern over "perfection" in SDI seems to revolve around this model (aside from the legitimate observation that there is no such thing as a leakproof defense). Arguments have raged over whether software can be adaptable enough to deal with unforseen attack strategies, and so forth.

I think that if automatic systems of that sort were advisable or achievable, we could phase out air traffic controllers, and leave the job to computers. Wars, even technological ones, will still be fought by men, with computers acting to coordinate communications, acquire and analyze target data, and control the mechanics of weapons system control. These tasks are formidable, and I make no judgement on which are achievable, and within what limits.

Both sides of the SDI debate have tended to use unrealistic models of technological warfare, the proponents to sell their program, the opponents to brand it as unachievable. The dialogue would be better served by agreeing on a model, or set of models, and debating the feasability of software systems for implementing them.

Dan Frank, uucp: ... uwvax!prairie!dan, arpa: dan%caseus@spool.wisc.edu

---

SDI impossibility

David Chase <rbbb@rice.edu>
Sat, 25 Oct 86 13:54:36 CDT

I don't know terribly much about the physics involved, and I am not convinced that it is impossible to build a system that will shoot down most of the incoming missiles (or seem likely enough to do so that the enemy is less likely to try an attack, which is effective), but people seem to forget another thing; SDI should ONLY shoot down incoming missiles. This system has to tread the fine line between not missing missiles and not hitting non-missiles.

I admit that we will have many more opportunities to evaluate its behavior on passenger airplanes, the moon, large meteors and lightning bolts than on incoming missiles, but we eventually have to let the thing go more or less on its own and hope that there are no disasters. How effective will it be on missiles once it has been programmed not to attack non-targets? To avoid disasters, it seems that we will have to publish its criteria for deciding between targets and non-targets (how much is an international incident worth? One vaporized weather satellite, maybe? If I were the other side, you can be sure that I would begin to try queer styles of launching my peaceful stuff to see how we responded).

I think solving both problems is what makes the software hard; it's easy to shoot everything if you have enough guns. We could always put truckloads of beach sand into low orbit.

David

Editorial on SDI

<decvax!utzoo!henry@ucbvax.Berkeley.EDU>
Fri, 24 Oct 86 00:31:56 edt

> ... The signatures were drawn from over 110 campuses in 41 states, and
> include 15 Nobel Laureates in Physics and Chemistry, and 57% of the
> combined faculties of the top 20 Physics departments in the country...

Hmmm. If a group of aerospace and laser engineers were to express an opinion on, say, the mass of the neutrino, physicists would ridicule them. But when Nobel Laureates in Physics and Chemistry express an opinion on a problem of engineering, well, *that's* impressive.

NONSENSE.

Dave Parnas, on the other hand, actually *is* an expert on the subject he has been expressing doubts about (the software problem). Although I'm not sure I agree with everything he says, I give his views a *lot* more credence than the people mentioned above.

Henry Spencer @ U of Toronto Zoology
{allegra,ihnp4,decvax,pyramid}!utzoo!henry

[I could have been a little more precise in my comment on Douglas Humphrey's message in RISKS-3.84. I said that Dave Parnas "does not PROVE that SDI is IMPOSSIBLE." By my curious emphasis, I meant to imply that Dave never even tried to prove impossibility. He said that the SDI software system would be untrustworthy. "..we will never be able to believe with any confidence that we have succeeded. We won't have any way of knowing whether or not SDI has succeeded."

Because Dave's comments really add significantly to this discussion -- and because Henry set me up -- let me quote an excerpt from a private note from Dave. PGN]

"SDIO's own report to congress quotes President Reagan about its goals. It says it is going to make nuclear weapons impotent and obsolete. They claim to be able to end the fear of nuclear weapons. They can do neither of these things unless they can make a trustworthy software system, one that we can rely upon. Without that, neither side will give up their offensive weapons.

"In short, the SDI software is not impossible, but ending the fear of nuclear weapons that way is." [David Parnas]

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 87

Sunday, 26 October 1986

Contents

- System Overload
  Mike McLaughlin
- Information Overload
  Mike McLaughlin
- SDI assumptions
  Herb Lin
- Info on RISKS (comp.risks)

System Overload

Mike McLaughlin <mikemcl@nrl-csr>
Sun, 26 Oct 86 21:13:56 est

Back in Systems 001 I was taught that an overloaded system, be it a reactor control or SDI, failed due to overload in the following manners:

1. Sacrificed quality of work.
2. Sacrificed throughput rate.
3. Failed catastrophically (crashed).
4. Any combination of the above.

Can a given system be designed to fail in a _chosen_ manner, so that it does not crash - i.e. "graceful degradation." Of course. I see no reason why new systems cannot do the same - at least in regard to the overload portion of the problem. - mikemcl@nrl-csr.arpa

---

Information Overload

Mike McLaughlin <mikemcl@nrl-csr>
Sun, 26 Oct 86 21:39:26 est

Undoubtedly we can load sensors on a system until it will no longer fly, move, fight, or whatever due to the number of sensors. Airplane cockpits already provide more information than pilots can handle. Combat sensor systems provide more data than battle-managers can handle. On the early space flights we even instrumented the astronauts themselves -- in a manner that should not be discussed on a family forum. There seems little point in providing a cockpit display of the pilot's rectal temperature; but on the ground someone cared.

One of the functions being performed by computers today is to filter the information, so that the system operator sees relevant data. One of the tough parts is to decide what is relevant. I submit that "operator assistant" computers deserve special care in design and testing. They seem to be used where lives are at stake, and where data is available. Relying on the computer to decide what is "relevant" in a given situation is fraught with risk. Relying on a human to decide in advance of the situation is not much better.

Another area of concern is the "transition" problem discussed in previous issues. I don't know that Navy Propulsion reactors are under-computerized deliberately, accidentally or at all. Having been a watch officer in the Navy and having lived through a number of unexpected emergencies I can personally attest to the seriousness of the "transition" problem - even without computers. To be awakened from sleep with alarm bells ringing and bullhorns blaring "FIRE, FIRE, FIRE IN NUMBER TWO MAGAZINE!" - and then be standing dressed, over the magazine, and in charge of the situation in less than 60 seconds is quite an experience. That I am here to recognize the problem is due to excellent train- ing of the entire crew, not to any specific actions on my part. Frankly, I just "went automatic" and shook after it was over, not during. I suspect that any pilot, truck driver, policeman, etc. could tell a dozen similar tales.

I'm not proposing any answers - except for extreme care.

- mikemcl@nrl-csr.arpa

SDI assumptions

<LIN@XX.LCS.MIT.EDU>
Sun, 26 Oct 1986 23:48 EST

> From: prairie!dan at rsch.wisc.edu (Daniel M. Frank)
>
> Much of the concern over "perfection" in SDI seems to revolve around this model (aside from the legitimate observation that there is no such thing as a leakproof defense).

I've said it before, but it bears repeating; no critic has ever said SDI software must be perfect. The only ones who say this are the pro-SDI people who are criticizing the critics.

> The [SDI] dialogue would be better served by agreeing on a model, or set of models, and debating the feasability of software systems for implementing them.

Having a "set of models" means that those models share certain characteristics. There is one major characteristic that all SDI software will share: we will never be able to test SDI software -- whatever its precise nature -- under realistic conditions. Then the relevant question is "What can we infer about software that cannot be tested under realistic conditions?"

Search RISKS using swish-e

Report problems with the web pages to the maintainer

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 88

Monday, 27 October 1986

Contents

- SDI, Missing engines, feeping creatureism in consumer products
  Roy Smith
- More aircraft instrumentation
  John Allred
- Re: Military vs. civilian automatic control systems
  Eugene Miya
- Perfection
  Douglas Humphrey
- Shipboard anecdotes
  Mike McLaughlin
- RISKS UNDIGESTIFIER on UNIX
  John Romine
- Info on RISKS (comp.risks)

SDI, Missing engines, feeping creatureism in consumer products

Roy Smith <cmcl2!phri!roy@seismo.CSS.GOV>
Thu, 23 Oct 86 15:28:25 edt

This message is a potpouri of several random thoughts that I've had in the past few days. The first two are apropos to recent topics on RISKS, the last is new material.

Re: SDI and unexpected inputs. I have a friend who works for the Army Night Vision Lab (I'm not sure that's actually the correct name). They work on "find the tank in the jungle at night" problems. He once described a program that looks for tanks in a battlefield -- the first thing it does is find the horizon and concentrate on the area below (i.e. the ground). My first thought was "what happens when they start dropping tanks by parachute?"

Re: Planes loosing engines. I gather than in many of the cases of planes having gross defects (i.e. a control surface torn off), the situation was at least meta-stable until the pilot tried to do something (i.e. turned off the auto-pilot to take control). I'm just guessing, but it seems that a chase plane could take off and intercept the damaged plane to make a visual inspection of its exterior quickly enough to be of some use. Am I being naive to think that this would be 1) practical and 2) of any use? Is it done already?

Re: feeping creatureism. There is an annoying trend towards computerizing things that just don't need computerization. Even worse is the urge to make things *seem* computerized when the microprocessor in them does nothing more than scan for switch closures on the control panel and run a simple timer. I recently bought an air conditioner -- it doesn't have a control panel, it has a "command center". It has the same controls (on/off, etc) as any other air-conditioner, but the panel is made up to look like some sort of computerized gizmo. My new electric dryer is the same way -- it's got "electronic drying", which means is it has a thermostat is the exhaust vent just like my mother's old mechanical-timer model. Speaking of my mother, she just bought a new car and hasn't figured out how the radio works yet because the familiar volume and tuning knobs aren't there any more.

So, how does all this tie in with COMPUTER RISKS? Take the dryer; by making it appear that there is some kind of computerized system monitoring and controlling the drying process, the consumer is duped into believing that his dryer is somehow better than the old ones. He doesn't really understand *why* it is better, but since it computerized, it *must* be better, right? Likewise with the car radio. While it may be true that digitally synthesized tuning is better than mechanical variable capacitors, (let's not start arguing about *that*) there was nothing wrong with the user interface (2 knobs to turn, maybe some pre-set pushbuttons). While the real advantage of the new radio over the old is the PLL instead of the variable cap, the *percieved* advantage is the "tune-up/tune-down" buttons instead of the tuning knob to turn. In fact, the new-fangled user interface is no better than the old one, and may in fact be worse.

Roy Smith, {allegra,philabs}!phri!roy
System Administrator, Public Health Research Institute
455 First Avenue, New York, NY 10016

more aircraft instrumentation

John Allred <jallred@labs-b.bbn.com>
Mon, 27 Oct 86 10:35:39 EST

Doug Humphrey asks:

" ... I am not sure why a pilot would need a video monitor to tell him that Number 2 just fell off the wing, ... He will no doubt understand this by the way the aircraft is acting."

A perfect example of why a pilot could use a monitor is the American Airline DC-10 crash at O'hare. The pilots knew they had lost power on the engine. However, they had no way of knowing that they had physically lost the engine (because you can't see the engines from the DC-10 cockpit.) Upon detecting that they had lost power in one engine, the pilots went exactly by the book - they changed the airspeed to best-2-engine-climb speed. Unfortunately, when the engine fell off the wing, it also ripped out some hydraulic lines in the wing, which were holding the slats (high lift devices on the leading edge of the wing) extended. With the slats retracted, the stall speed of the damaged wing was *above* best-2-engine-climb speed. So, one wing stalled, the other kept generating lift, and the plane rolled over.

It should also be noted that pilots in simulators, when given the exact same situation, were able to save the aircraft when they knew that they had physically lost the engine, while pilots that did not know uniformly failed to save the aircraft.

Doug is correct in stating that a pilot should be able to understand if he's lost something important. However, that understanding could come too late, or in and of itself be fatal.

Re: Military vs. civilian automatic control systems

Eugene Miya <eugene@AMES-NAS.ARPA>
Mon, 27 Oct 86 09:04:33 pst

I basically agree with Will's thesis about missions, but I don't the difference is that simple (binary). Two years ago, an F-8 Crusader (single engine Navy fighter, older) lost power over San Diego. The pilot had time to eject, but before doing so, he tried to avoid hitting buildings in the Serrento Valley area. (True he might have misjudged prior to ejection, but the plane did come down in a parking lot and not the nearby electronics buildings.) Many pilots have faced this dilemma in the past: including civilian pilots (do I kill several hundred people on the ground in addition to the passengers I have just killed?). I think this also goes for civilian rescue missions. Ford' Mayeguez (sp) mission in 1975 cost more Marine lives than civilians rescued. True we will never know the real political consequences of not rescueing (liberals: "we would have negiotated release," conservatives: "they would have died"), but my point is many of the fundamental types of systems are no different in the civilian or military sphere, and that there is overlap (with tricky trade offs) with military operations.

--eugene miya

Perfection

Douglas Humphrey <deh@eneevax.umd.edu>
Mon, 27 Oct 86 02:52:25 EST

To LIN : In response to a message, you state that none of the anti-SDI
folk ever stated that the software had to be perfect. I have
heard constantly in both the widely read (Washington Post) and
limited (?) distribution industry media (Aviation Leak and

Space Mythology) SDI critics that contect that it must be perfect
or it is useless. I don't beleive this, and I would hope you
don't either, but saying that the whole must be perfect
certainly implies that the parts must be perfect. (Opps. contend..)

About failure modes in software systems, yes, it is possible to
design fault tolerant and fault permissive systems. Systems that
have a know 'prefered failure mode'. Example, hardened underground
facilities, I have been told (no references here) are not designed
to withstand forces equaly throughout the structure. That would mean
that when the structure finaly failed under load, there would be no
reliable way to project where the failure would happen. Better to
design with structural over load failure in mind and specificaly
designate one area as the failure area, and then take withever
measures one can (air/water tight bulkheads, etc.) in that
area since you now have a high degree of confidence that the failure
will happen where you want it, and are ready for it.
Software can be designed the same way by dealing not only with the
quantity (targets) by the quality of targets (destinations) and
selectivly 'failing' on those which are the least important.

I would guess that a catostrophic failure would be the one
to avoid, even of the system decided that it was time to reboot,
clearing target tracking data since some of it was detected as
bad. The system might then let through whatever was locked at the
time of the failure, but at least it would resume defense rather than
either crash outright, or get into a position where its target load
started to effect its real time processing and maybe preventing it
from reacting well enough to to its job.

Hey ! If we get flaming about this much deeper, we should all start
submitting bills to SDIO.......

Doug

Shipboard anecdotes [marginally relevant but intersting]

Mike McLaughlin <mikemcl@nrl-csr>
Mon, 27 Oct 86 13:05:11 est

Two anecdotes about shipboard emergencies.
In that fire, one sailor did think about what was happening, and ran aft as fast as his little legs would carry him. A _giant_ Chief Gunner's Mate named Mills grabbed him, pointed him back to his battle station, and said something like "Son, you better get to your battlestation. When a destroyer has a fire in a magazine, you just can't run far enough!"

In another emergency that was really too complex to explain on Risks, I _really_ went automatic. I had far more charge of the situation, and far more depended on my own actions. Simply put, the USS Saratoga was about to run over us, and we had lost control of our rudders. I did the requisite things, and am here to tell about it. But _during_ the experience

I was "out of body" - Some part of me was floating above and behind me, watching me give orders & do things, sort of supervising/monitoring me, but not interfering. I have no recollection of the situation from my body's eyes and ears once the situation developed. All of my quite detailed memory is from that viewpoint floating up in the aft port quarter of the pilothouse. I must have done good, because everybody said so, from the skipper down to the real authorities, the mess cooks. I have to conclude that I had been so thoroughly trained that I was operating on a learned-reflex basis, leaving my conscious mind free to observe. I don't know if we can use that somehow in designing "operator assistants" or not.

- Mike

RISKS UNDIGESTIFIER

John Romine <jromine@nrtc-gremlin>
Mon, 27 Oct 86 10:24:41 -0800

If you have the MH Message Handler (a user agent for UNIX) you have the "burst" command which seems to work just fine on Risks digests. MH is now distributed as user-contributed software on the 4.3BSD tape, and is available for anonymous ftp from the host louie.udel.edu. Also, you can get a magtape copy for $75 from the University of California, Irvine. I've included the release announcement below.

/JLR

A new release of the UCI version of the Rand Message Handling (MH) system is available for distribution. This release of MH is called

MH 6.5

There are a lot of changes between MH.6 and MH 6.5; a lot of performance enhancements were made, there's also a lot of support for distributed mail (personal mail and bulletin bboards).

Here are the details:

- MH is in the public-domain
- MH runs on a number of versions of UNIX (4.[123]BSD, V7, SYS5, and related variants, e.g., HPUX) [sorry, no support for SYS3.]
- MH runs on top of a number of mail transport systems (MMDF-{I,II}, SendMail, stand-alone (with UUCP support))

Although MH is not "supported" per se, it does have a bug-reporting address, Bug-MH@ICS.UCI.EDU. Bug reports (and fixes) are welcome, by the way. There are also two ARPA Internet discussion groups: MH-Users@ICS.UCI.EDU and MH-Workers@ICS.UCI.EDU (somewhat analogous in charter to Info-UNIX and UNIX-Wizards).

There are two ways to get a distribution:

1. If you can FTP to the ARPA Internet, use anonymous FTP to
louie.udel.edu [10.0.0.96] and retrieve the file portal/mh-6.tar.
This is a tar image (approx 4MB). The file portal/mh-6.tar.C is
the tar image after being run through the compact program (approx
2.3MB). The file portal/mh-6.tar.Z is the tar image after being run
through the compress program (approx 1.5MB).

2. You can send $75 to the address below. This covers the cost of
a magtape, handling, and shipping. In addition, you'll get a
laser-printed hard-copy of the entire MH documentation set. Be sure
to include your USPS address with your check. Checks should be made
payable to

Regents of the University of California

and must be drawn on U.S. funds. It's also a good idea (though not
mandatory) to send a computer mail message to "Bug-MH@ICS.UCI.EDU" when
you send your check via USPS to ensure minimal turn-around time.
The distribution address is:

Support Group
Attn: MH distribution
Department of Information and Computer Science
University of California, Irvine
Irvine, CA 92717

714/856-7553

Sadly, if you just want the hard-copies of the documentation, you
still have to pay the $75.00. The tar image has the documentation
source (the manual is in roff format, but the rest are in TeX
format).

/mtr

Search RISKS using swish-e

Report problems with the web pages to the maintainer

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 89

Tuesday, 28 October 1986

Contents

- Airplanes and risks
  Alan Wexelblat
- TSE, Air Canada
  Matthew Kruk
- Big Bang
  Robert Stroud
- Physicists on SDI and engineering..
  Herb Lin
- ABM, SDI, and Freeman Dyson
  Peter Denning
- Info on RISKS (comp.risks)

Airplanes and risks

Alan Wexelblat <wex@mcc.com>
Tue, 28 Oct 86 11:23:52 CST

Today's paper has a couple of airplane-related items that got me to thinking.

One item is a story on how the FAA is going to adopt strict rules for small aircraft in busy airspaces and establish a system to find an punish pilots who violate these rules. The question this brought to mind is: is this the right approach for the FAA's problem? How about for computer systems? Can (or should) we manipulate the user so that he uses the system the way we designers intended it to be used? Is training the answer (as suggested by the Navy emergency stories)?

The next item is an analysis of the emergency aboard the Thai jet. Apparently the fault is similar to the one that doomed the JAL 747 that crashed recently in Japan. The factor that made the difference -- according to Hiroshi Fujiwara who is deputy chief investigator of Japan's Aviation Accident Investigation Commission -- was that the Thai Airbus A-300 retained hydraulic control of the flaps and rudder on the tail.

Both the 747 and the A-300 have triply-redundant hydraulic systems, but on the 747 all three pass through the rear bulkhead in the same opening. Thus all three were ruptured at once. On the A-300 there are three separate openings and while two of the systems were ruptured in the Thai jet, the third remained usable.

The related question is: can we make use of this feature in computer systems (hardware or software)? That is, if a program has three ways of doing something can we isolate them so that a bug somewhere doesn't simultaneously cripple all three? Can we (given needs like security) separate computer hardware so that it is much more difficult to simultaneously destroy primary and backup hardware?

Comments and discussion welcomed.

Alan Wexelblat
ARPA: WEX@MCC.ARPA or WEX@MCC.COM
UUCP: {seismo, harvard, gatech, pyramid, &c.}!ut-sally!im4u!milano!wex

---

TSE, Air Canada

<Matthew_Kruk%UBC.MAILNET@MIT-MULTICS.ARPA>
Mon, 27 Oct 86 10:46:30 PST

No doubt you will hear more about these items from better informed sources. I merely heard brief summaries on the morning news today (Monday, 27th).

1. The Toronto Stock Exchange computer went down for about 5 minutes this morning. No cause given (yet).

2. A fire in a building, which houses the main computer (reservations?) of Air Canada, in Montreal. An Air Canada official cannot predict the effect on people holding advance registration. Damage cost estimates run in the millions.

Presumably there will be more information in tonight's paper. I'll try to get a summary out as soon as I can.

Big Bang [Also noted by Martin Minow. Thanks.]

Robert Stroud <robert%kelpie.newcastle.ac.uk@Cs.Ucl.AC.UK>
Tue, 28 Oct 86 19:42:40 gmt

Yesterday, October 27th, was the day of the Big Bang in the City - a revolution in the way in which the Stock Exchange is organised. Basically, three things happened - the market was opened to foreigners, the distinction between jobbers (who trade on their own account) and brokers (who buy and sell on behalf of clients) was abolished (thereby introducing potential conflicts of interest and necessitating the erection of so-called Chinese Walls to prevent this), and finally, guaranteed minimum commissions were removed, making things much more competitive. Wall Street went through something like this on May Day a few years ago.

Anyway, these three changes led to the introduction of new computing systems developed in something of a rush to meet yesterday's deadline. Most important of these was the Stock Exchange Automated Quotation system (SEAQ) which several companies had to switch to by default at the last minute when they realised that their in-house systems would not be working in time. SEAQ provides information over the Topic network to 10,000 terminals about share prices - dealing is still done manually (at least until next year) although the SEAQ system is supposed to be updated continuously to reflect the trading.

There was a full-scale rehearsal last week when the Stock Exchange opened on a Saturday for the first time in its history. Not everything went smoothly and there were complaints about prices not being updated for as long as 20 minutes, making it possible to buy at one price and simultaneously sell at another. However, as late as Sunday afternoon, the chairman of the Stock Exchange Council was defiantly challenging anyone to demonstrate that this was still a problem.

Well, I'm sure that RISKS readers can guess what happened on Monday morning. The system lasted half an hour before it broke down at 8.30am! Although it was later up and running, and the problem was with the antiquated Topic network rather than the SEAQ system itself, there are fears that it could happen again under crisis. Apparently, this failure was caused by curiosity - everybody wanted to try out the new system at once, and it couldn't cope.

Curiosity is an interesting example of human behaviour causing a computer system to fail. I believe the telephone companies have a similar problem on Mother's Day when the pattern of usage is abnormal.

Another example of human behaviour has been the reaction of the dealers to the new system, to some extent invalidating the whole concept. Only time will tell whether this is just suspicion of a new technology or a real problem. However, at present the dealers are rather wary and are therefore only offering small deals on the system (up to 1000 shares) so that the big deals (100,000) are still negotiated over the telephone. This is partly a defensive move because the system is (rightly or wrongly) perceived as being slow, making it possible to offer unrealistic prices not in line with the market - the real market is off the screen. Equally, some market makers "are playing complicated games to test their competitors and this is likely to become a feature of the new markets". One dealer has even gone so far as to describe the SEAQ terminals as "useless". [This paragraph extracted from an interesting article in today's Times entitled "New screens 'fail to catch full deals'" by Richard Thomas]

Naturally, there has been a wealth of material about all this in the media recently, and today, all the papers are competing with each other for puns on Big Bang! When the dust settles on this most public of failures, RISKS archaeologists will have plenty of relics to excavate. Here is one of the more technical articles, reproduced without permission from today's Times, (28th October p.21)

Robert Stroud,

Computing Laboratory,
University of Newcastle upon Tyne.

ARPA robert%cheviot.newcastle@ucl-cs.ARPA (or cs.ucl.ac.uk if you trust domains!)
UUCP ...!ukc!cheviot!robert

++

"Big Bang shambles as computer breaks down -
Goodison blames Topic subscriber's curiosity"

by Michael Clark

Yesterday's disastrous debut for the Stock Exchange Automatic Quotations system was a prime example of Murphy's Law: "If something can go wrong, it will". But the problems encountered by dealers on the trading floor stemmed from technical problems at Topic, the Stock Exchange's own tried-and-tested screen-based information system.

Topic went off the air at 8.30am - a crucial time for traders hoping to establish the price of stocks ahead of the official start of dealings at 9am - and stayed down for more than an hour, apart from one intermission. The break also resulted in all operations on SEAQ being suspended for the same period.

Stock Exchange officials blamed a breakdown in the link between Topic and SEAQ. Market-makers feed their prices into the SEAQ computer which are then updated and displayed on the 10,000 Topic terminals situated in the City offices of brokers and fund managers.

Sir Nicholas Goodison, chairman of the Stock Exchange Council, described Topic as the world's eye on the market and said that although it had enjoyed a high level of reliability, it was six years old and considered fairly antiquated by today's standards.

A Stock Exchange spokesman quickly blamed curiosity for the failure: "The system cannot handle all the Topic sets being used at the same time."

Topic was operating at maximum capacity yesterday, receiving 12,000 page requests a minute, or 200 per second. [SEAQ itself is designed to handle 40 transactions per second, but the maximum demand yesterday was 22 per second.] Sir Nicholas said that the system had suffered a small setback which had been put right. He said that Topic had been overwhelmed by the number of page changes which, normally, it would not have to cope with. Most of it was simply curiosity by subscribers.

"If you want to put a monkey, or a dodo in a zoo, everyone will want to look at it on the first day," he said.

But it is still possible the breakdown could happen again. SEAQ encourages dealers and fund managers to use its screens more and a sudden surge of business may overload Topic.

The Stock Exchange's technical officers say there are only a few adjustments that can be made to Topic. One may be to introduce an automatically triggered queuing system which limits the number of subscribers using the system at any one time. But many dealers fear this could lose them business.

Meanwhile, there were still complaints from market makers about the time it took for a price change to appear on Topic after dealing. There were reports of delays up to one hour. Sir Nicholas said these would be checked but still blamed market makers' own internal systems for the delay.

Physicists on SDI and engineering..

<LIN@XX.LCS.MIT.EDU>
Mon, 27 Oct 1986 20:01 EST

> From: decvax!utzoo!henry at ucbvax.Berkeley.EDU
>
> Hmmm. If a group of aerospace and laser engineers were to express an opinion on, say, the mass of the neutrino, physicists would ridicule them. But when Nobel Laureates in Physics and Chemistry express an opinion on a problem of engineering, well, *that's* impressive.

I simply point out that the Manhattan Project was run by a bunch of physicists. The H bomb was transformed from an 80 ton clunker to a practical device by physicists. These were "mere" engineering problems too.

ABM, SDI, and Freeman Dyson

Peter Denning <pjd@riacs.edu>
Tue, 28 Oct 86 11:10:29 pst

In RISKS 3.83, Ken Dymond noted that the ABM (anti ballistic missile system) debate of the early 1970s is similar to the SDI debate of the mid 1980s, and asked for sources that might shed light on the past debate. Here's one source known to me:

Chapter 7 in Freeman Dyson's WEAPONS AND HOPE is an excellent analysis of the ABM debate. He compares that debate with the ``star wars'' debate and finds both similarities and differences. He sees a role for (nonnuclear) ABM systems in a nuclear-free world, and expresses the hope that the ABM debate will one day be reopened. In contrast, he considers ``star wars'' a technical folly, for reasons having little to do with the reliability of the software systems.

Peter Denning

Search RISKS using swish-e

Report problems with the web pages to the maintainer

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 90

Thursday, 30 October 1986

Contents

- Anti Skid Brakes
  Paul Schauble
- The Mother's Day Myth, and "Old Reliable"
  Jerome H. Saltzer
- Collision avoidance systems
  John Larson
- Crime and punishment
  Peter Ladkin
- Air Canada
  Matthew Kruk
- (Voting) Machine Politics
  Mike McLaughlin
- Computer RISKS in "Ticker-Tape Parades"
  PGN
- SDI vs. Social Security
  Scott Guthery
- SDI Impossibility?
  Scott Dorsey
- Feeping Creaturism
  Charley Wingate
- Info on RISKS (comp.risks)

Anti Skid Brakes

Paul Schauble <Schauble@MIT-MULTICS.ARPA>
Thu, 30 Oct 86 04:44 EST

In view of the recent discussion on Anti-Skid Brakes and their overrides, I thought I would post this item. It is by John Dinkel's column in the October 1986 issue of Road & Track and describes his and other race drivers experience with the Anti-skid Braking System (ABS) on a Corvette.

\- - - - - - - - - - - - - - - - - - -

During a recent test session at Sears Point International Raceway, the Bakeracing Corvette drivers were treated to a couple of graphic demonstrations of the differences between ABS and non-ABS braking. Coming down the Carousel, a long sweeping, downhill left-hander, team leader Kim Baker found himself running a bit fast for the wet track conditions. Rather than drive off the track, Kim locked the brakes and put the car into a harmless spin. Surprise. This time it wasn't totally harmless. Once the car stopped sliding sideways, the ABS caused the Vette to steer in the direction in which the front wheels were aimed. In this instance the ABS allowed the car to take a wider than expected arc, and Kim and the Corvette found themselves rolling gently into the tire wall on the outside of the turn. No harm except for embarrassment on Kim's part, but this incident certainly pointed out one of the differences between spinning a car with and without ABS.

That wasn't the only difference. I listened intently as two of out drivers complained of lack of braking and a soft pedal as they applied the brakes at the top of the Carousel. Having just finished driving several laps following a discussion with John Powell, owner of one of the other Corvette teams and an experienced driver training instructor, about ABS versus non-ABS race track driving, I knew what the problem was. Coming up to the braking point at the entrance to the Carousel, a car gets light as it crests a hill. If you apply ABS brakes at the instant, the ABS senses loss of traction or a low-coefficient [of friction] surface and releases pressure to one or more wheels that it thinks is trying to lock. The ABS brain has been fooled by the car losing download over that crest, and it can take up to half a second for the system to recover and allow full braking force after the wheel loads return to normal. What does the driver sense during that half second besides panic? A soft pedal and longer than expected braking distances. The solution? Simple. Initiate your braking right before the car gets light or wait until the wheels are fully loaded again after the crest. Exercise either of these two options and you'll never know that the car is equipped with ABS except for the added security it affords when you hot foot it into a corner and discover that you can still steer into the turn despite having the brakes "locked". And, as we discovered at Portland, a Corvette with ABS and drive rings around the competition on a wet track.

- - - - - - - - - - - - - - - - - - - -

It's noteworthy that some racing teams are experimenting with computer controlled cars. The suspension, braking, steering, and engine parameters under direct control of an on-board computer that is programmed for the specific race and track being driven. So far, such a car has not run in competition. However, as Risks readers know, computer controlled engines and transmissions are almost commonplace. I expect to see the car with the computer controlled suspension in competition in 1987.

The Mother's Day Myth, and "Old Reliable"

Jerome H. Saltzer <Saltzer@ATHENA.MIT.EDU>
Tue, 28 Oct 86 23:11:16 EST

From Robert Stroud's piece on SEAQ. . . (RISKS-3.89)

> Curiosity is an interesting example of human behaviour causing a
> computer system to fail. I believe the telephone companies have a
> similar problem on Mother's Day when the pattern of usage is abnormal.

Workers in the New England Toll Switching Center here in Cambridge tell visitors on guided tours (that is the best I can do for a reference; sorry) that their busiest day for long distance calls is the Monday after Thanksgiving. The explanation they give is that the Friday after Thanksgiving is the first real Christmas shopping day, because so many people have or take that day off. All the retailers in New England study the pattern of sales on Friday and Saturday, ponder it on Sunday, and spend Monday morning on the telephone to their suppliers trying frantically to get their hands on more of whatever seems to be selling well this year.

That one falls in the category of hard-to-imagine-in-advance-but-easy-to-explain-in-retrospect system problems.

The Michael Clark article quoted by Stroud contains a comment that is eyebrow-raising from the point of view of RISKS:

> . . . said that although it had enjoyed a high level of reliability,
> it was six years old and considered fairly antiquated by today's
> standards.

I wonder who it is that considers that system as antiquated? Another perspective says that a complex system that has been running for six years is just beginning to be seasoned enough that its users can have some confidence in it. People who have work to do (as compared with computer scientists, who users perceive as mostly interfering with people trying to get work done) know that in many cases the most effective system is one that has just become obsolete. The tinkerers move on to the shiny new system and leave the old one alone; it becomes extraordinarily stable and its customers usually love it.

Jerry Saltzer

Collision avoidance systems

<jlarson.pa@Xerox.COM>
Wed, 29 Oct 86 11:29:49 PST

There was a rather distressing article about collision avoidance systems in the San Jose Mercury News recently (Sun, 26 Oct). According to the article the FAA nixed a workable collision avoidance system designed by Honeywell 11 years ago because it competed with an in house collision avoidance system they were developing. This was done in spite of several studies showing that the Honeywell system would be better than the FAA system. The Honeywell system would have cost $14,000 per comercial airline and was projected to be cost reduced to about $1000 making it affordable for most aircraft.

They also quoted a former FAA official to the effect that the FAA was partly responsible for the loss of over 700 lives due to collisions because of their failure to go ahead with the Honeywell system.

The FAA is finally almost ready with their own version of a collision avoidance system (apparently needs another year of testing), but it will cost a lot more than the original Honeywell system ($40-70K) and has problems with clouds and bad weather. It also apparently can't be made as cheap as the original Honeywell system ($5,000 or so) so it will probably not be used much except in commercial aircraft.

Does anyone know more about this issue ? I'm particulary interested in technical details about the Honeywell and the FAA systems.

John

Crime and punishment

Peter Ladkin <ladkin@kestrel.ARPA>
Tue, 28 Oct 86 18:34:59 pst

Alan Wexelblatt asks:

[...] the FAA is going to adopt strict rules for small aircraft in busy airspaces and establish a system to find and punish pilots who violate these rules. The question this brought to mind is: is this the right approach for the FAA's problem?

These rules are already in existence, and so are the punitive practices. Neither can stop mistakes, as in the Cerritos airspace violation by the Archer. They are even less effective against deliberate violators, who turn off their transponders.

How about for computer systems? [..] Is training the answer [..] ?

Maybe to avoid mistakes, as in rm *, but not for deliberate violators.
The late-70s Berkeley Unix cracker was known, and wouldn't stop.
I believe that the Computer Center tried to hire him to turn his talents to useful purposes - which didn't work.
Eventually the police went around to arrest him, which seemed to work (he was a young middle-class teenager).
So training wasn't the answer, but sufficiently severe punishment was, in this case. Not that I advocate this approach.

Peter Ladkin

Air Canada

<Matthew_Kruk%UBC.MAILNET@umix.cc.umich.edu>
Wed, 29 Oct 86 08:37:58 PST

Apparently this was not the main computer system but a (reservations) backup system. The "stupidity" of this situation is that, according to news reports, major building damage (currently estimated at greater than $10 million) might have been avoided had their been a sprinkler system. I would be interested in knowing how it came to be decided to have a "backup system" located in such a building and if there was additional data security measures were taken by Air Canada (initial newspaper reports seem to imply that there were none). Perhaps Risks readers in eastern Canada might be able to shed more light on this.

(Voting) Machine Politics

Mike McLaughlin <mikemcl@nrl-csr>
Wed, 29 Oct 86 16:11:13 est

See DATAMATION, 1 Nov 86, Vol 32 No 21, "Machine Politics" beginning on page 54. Good article by John W. Verity. Quotes Deloris J. Davisson of Emerald Software & Consulting, Inc., Terre Haute, Ind., and of Ancilla Domini College. If anyone knows Ms. Davisson, request she be invited to contribute to Risks.

Computer RISKS in "Ticker-Tape Parades"

Peter G. Neumann <Neumann@CSL.SRI.COM>
Thu 30 Oct 86 03:01:32-PST

Mets fans were treated to an interesting new form of computer risk on Tuesday. An estimated 2.2 million people turned out for the parade to honor the Mets, so clearly more paper had to be found to dump on the people in keeping with New York's tradition of a ticker-tape parade. The solution was to use computer printout as well as ticker-tape, including huge volumes of billing reports, shipping orders, and stock records. Thus, we ask our New York RISKers whether they picked up any interesting print-out that might have been a violation of privacy. Scavenging dumpsters is an old art, but having possibly sensitive printouts raining down on you is a new twist.

<"guthery%ascvx5.asc@slb-test.CSNET">
Tue, 28 Oct 86 07:37 EDT

<"4596::GUTHERY%slb-test.csnet"@CSNET-RELAY.ARPA>
To: risks@CSL.SRI.COM
Subject: SDI vs. Social Security

When I think about the risks of computerization, I'm much more afraid of the Social Security System than I am of SDI. We know computers hitched to things-that-go-boom are dangerous so we watch them carefully as we build them and as we use them. But computers hitched to paper? Who really cares? If it issues a check that's too small or a report that's fallacious, it's the recipient's problem to make it right. Right?

In other words, if the builders and maintainers of the system have vested interests in the correctness of the system it is more likely to be correct than if they don't. Said another way, it is always the "users" who are ultimately in charge of ... not responsible for, mind you ... debugging the program. Things get fun when the only means a "user" has to debug the system is a bureaucratic hole to yell into.

But beyond these mild inconveniences to that lowest of all computer life, there is a more ominous shadow on the horizon. We are bringing into being very large systems whose behavior we don't understand yet which are woven into the fabric of our daily life. I don't mean we don't understand the line that says multiply hours worked by hourly pay. I mean we aren't in control of it or its destiny. We can't describe its global behavior. We change it but we don't know where its evolutionary path is leading.

("Well, son, it started out as a computer program but we just kinda lost track of it. Now it's kinda like the law of gravity. We take it as given and just try to work with it or work around it.")

What do we know about scaling up and evolving software? Are there any empirical studies of the evolution of large code bodies (5+ million lines, 10+ years)? Do we know how to engineer global behavior from local function? How do we recover functional descriptions and domain-specific knowledge from large, mature software systems?

Software productivity always seems to mean bringing more code into being quickly. Yet the problem I fear is that there is too much code of unknown quality and function scattered everywhere and then forgotten.

I suggest that we already have many of the problems that the SDI critics call out ... only in a more innocuous form. Cancer kills just as surely as a bullet but it's a hell of lot harder death. We all seem to be sitting around smoking cigarettes and worrying about being shot.

SDI Impossibility?

Scott Dorsey <kludge%gitpyr%gatech.csnet@CSNET-RELAY.ARPA>
Mon, 27 Oct 86 18:36:49 est

> "In short, the SDI software is not impossible, but ending the
> fear of nuclear weapons that way is." [David Parnas] (RISKS-3.86)

Is such reliable software impossible? In 1967, a conference on computer systems in space contained a paper certifying that the software required for the Apollo missions was so complex and hard to certify that

it would never work. Maybe at the time it was true. And it was certainly
true that it did not work the first time. The point that I am making is
that no one can really forsee how far software engineering technology will
advance in the next few years, and how far simulation technology will
advance. Is it worth spending money for something that may not work?
In my (* opinion *) it is always worth spending money on pure research,
but my position is a bit biased.

Scott Dorsey
ICS Programming Lab (Where old terminals go to die), Rich 110,
Georgia Institute of Technology, Box 36681, Atlanta, Georgia 30332
...!{akgua,allegra,amd,hplabs,ihnp4,seismo,ut-ngp}!gatech!gitpyr!kludge

Feeping Creaturism

Charley Wingate <mangoe@mimsy.umd.edu>
Tue, 28 Oct 86 22:42:02 EST

(Follow-up to Roy Smith)

This gratuitous computerization also has the obvious risk of introducing a
useless level of unreliability in the system without much gain in
performance. This is especially a problem for consumer products, where the
electronics are in a far from ideal environment, and which are modularized
to the point of guaranteeing a world tantalum shortage in the not-too-distant
future :-).

C. Wingate

Search RISKS using swish-e

THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 3: Issue 91

Thursday, 30 October 1986

Contents

- Evolution, Progress
  Jim Horning
- System Overload
  David Parnas
- "Perfect" systems from imperfect parts
  Bob Estell
- The software that worked too well
  Dave Benson
- Assessing system effectiveness
  Dave Benson
- Risks of raining computer print-out
  Alan Wexelblat
  Martin Ewing
  PGN
- Info on RISKS (comp.risks)

Evolution, Progress

Jim Horning <horning@src.DEC.COM>
Thu, 30 Oct 86 15:15:51 pst

"guthery%ascvx5.asc@slb-test.CSNET" asks:

> What do we know about scaling up and evolving software? Are there any empirical studies of the evolution of large code bodies (5+ million lines, 10+ years)? Do we know how to engineer global behavior from local function? How do we recover functional descriptions and domain-specific knowledge from large, mature software systems?

There have been at least a few such studies. The one I can retrieve most quickly is "Programs, Cities, Students--Limits to Growth?" reprinted in PROGRAMMING METHODOLOGY: A COLLECTION OF ARTICLES BY MEMBERS OF IFIP WG 2.3, Edited by David Gries, Springer-Verlag, 1978. Belady and Lehman published a number of other articles based on their studies of the metadynamics of systems in maintenance and growth. (Their studies are to most studies of programming as Thermodynamics is to Classical Mechanics: They stand back far enough that the activities of individual programmers can be treated statistically.)

Scott Dorsey comments

> that no one can really forsee how far software engineering technology will advance in the next few years, and how far simulation technology will advance.

I agree, and certainly am in favor of research. However, the recent past is often a good predictor of the near future. A good measure of the progress of software engineering in the last 18 years is to compare the proceedings of the two NATO conferences in 1968 and 1969 with the contents of RISKS. The NATO proceedings were reprinted in SOFTWARE ENGINEERING: CONCEPTS AND TECHNIQUES, edited by J. M. Buxton, Peter Naur, and Brian Randell, Petrocelli/Charter, 1976. I think many people will be surprised and disappointed at how little the problems and approaches have changed in that time. I interpret this to mean that our ambitions for computer systems have grown at least as rapidly as our abilities to produce them.

Jim H.

System Overload (RISKS-3.87)

<parnas%qucis.BITNET@WISCVM.WISC.EDU>
Tue, 28 Oct 86 07:25:40 EST

Mike McLaughlin raises the interesting issue of system overload in software systems. I think RISKS readers should focus on that issue with regard to hard real-time systems, systems in which an answer too late is worthless. A long time scale example of a hard real-time system is a weather forecast. If you receive it after you have experienced the weather, it is of little value. A more familiar example is a bomb-release computation. If you are told, release 10 ms ago, the information is useless. Overload in such systems can make them useless. The only solution is to make sure that overload cannot happen. However, this is not the same as making sure that the system will not be aware of the overload.

According to the BSTJ articles on the ABM system known as SAFEGUARD, the system protected against overload by knowing its limits and refusing to attempt to deal with a new attacking missile if this would cause overload. This guaranteed capacity to handle the load that was being handled and meet the real-time deadlines.

The same approach is often used for handling overload in telephone switching. If calls exceed the capacity users are asked to wait. There are delays in getting a dial tone or in a call going through.

Clearly, there are differences in the two situations. In the telephone situation the callers wait, they have little choice and the delay, while it may be annoying is seldom critical. In the ABM situation, the missiles don't wait; they do not need the services of the defense system anyway.

In fact, the solution of ignoring newly arriving "users" gives rise to an effective countermeasure, send your decoys first. Thus, our inability to provide infinite capacity in real-time systems gives rise to an unavoidable weakness when dealing with an enemy. The finite limit is always there, and there are often cheap ways to exploit it. We should note that the same situation arises in a telephone system. I am told that when President Kennedy was shot, many Washington telephones did not respond because of overload. Rock concerts have been known to have similar effects. If you are planning a live version of "The Mouse that Roared" announce the availability of a large number of cheap tickets for a popular group or groups just as you attack.

There is a simple but important lesson here. There are clear limits on what we can do and in an adversary situation those limits can be exploited. Nobody would suggest that we should not have built the telephone system because of these inherent weaknesses, but we would laugh out loud if those who make their living by developing telephone systems were to advertise a system that could not be defeated by a determined and sophisticated enemy.

"Perfect" systems from imperfect parts

"ESTELL ROBERT G" <estell@nwc-143b.ARPA>
30 Oct 86 13:47:00 PST

Did I *really* read in a recent RISKS that for a system to work perfectly, each component in it must work perfectly?

Well, if by "perfect" one means no errors anywhere, no matter how minor; and if by "system" one means a collection of parts connected in series; then I guess I agree.

But if "perfect" can be defined as "don't let any runs score" then the recent World Series offers a counter-example. The Mets got hits in game #1; there were base runners - just none of them got all the way around to score. What's more, balls that got by one infielder were scooped up by another, with the result that the batter was still thrown out.

It's been a long time now since I had to rely on a computer system that was a single thread series of non-reduntant parts; our systems do have troubles; memory modules fail; CPU's fail; mag tapes and disks and printers fail; communications lines, and modems fail. So the system comes up [stays up?] in degraded mode; I get my work done.

Maybe we should abandon the debate about SDI, and just roll up our sleeves and make something work acceptably. Doesn't have to be high energy beam; probably should not be space based. Undoubtedly should be a collage of over-lapping and co-operating subsystems. Those subsystems that get done first can be deployed first; maybe some off-the-shelf technology is ready now. Some of the subsystems can be used against targets other than ICBM's;

e.g., cruise missile defenses might also work well against drug runners.

The RISK I'm beginning to see is that if we who know well enough how to design redundant systems don't help, others may design SDI as a "chain no better than it's weakest link." If they do: (a) The links will be VERY strong, and (b) gold-plated - so they won't corrode; and (c) it will cost way too much; and (d) it still won't work.

RGE
Opinions expressed are entirely personal.

The software that worked too well

Dave Benson <benson%wsu.csnet@CSNET-RELAY.ARPA>
Wed, 29 Oct 86 17:31:59 pst

This story is nth hand, thus to be classified as rumor. But it is relevant to RISKS, so I pass it on, if only as a parable.

SeaTac is the main Seattle-area airport. Ordinarily aircraft landings are from the north, and this end of the runway is equipped with all the sensing equipment necessary to do ALS (Automatic Landing System) approaches.

The early 747 ALS worked beautifully, and the first of these multi-centaton aircraft set down exactly at the spot in the center of the runway that the ALS was heading for. The second 747 set down there. The third 747 landed on this part of the runway. ... As did all the others.

After a while, SeaTac personnel noticed that the concrete at this point at the north end of the ALS runway was breaking up under the repeated impact of 747 landings. So the sofware was modified so that 3 miles out on the approach, a random number generator is consulted to choose a landing spot -- a little long, a little short, a little to the left or a little to the right.

THE MORAL:
Don't assume you understand the universe without actually experimenting.

Assessing system effectiveness

Dave Benson <benson%wsu.csnet@CSNET-RELAY.ARPA>
Wed, 29 Oct 86 17:31:42 pst

(sp == Scott Preese)
sp> Dave Benson argues that it is more reasonable and conservative to assume
sp> that an overloaded system will fail entirely than to assume it will either
sp> perform at its design limit but no more or perform above its design limit.

sp> That's unarguably the conservative assumption. I would deny that ANY
sp> assumption was reasonable, given only a performance ceiling and the
sp> knowledge that performance demand will exceed that ceiling.

Might be helpful to look to the history of engineered artifacts, especially military artifacts, and most especially military software artifacts. Then your "givens" are no longer the only data to bring to bear on the problem.

sp> It is obvious that the system could be designed to perform in any of
sp> the suggested ways when unable to cope with load.

While it might be possible to DESIGN the system to perform in any of a number of ways, there is no particularly good reason to believe that a software system would, in fact, meet those design goals. There is plenty of evidence to suggest that military software can only meet design goals after repeated operational testing and rework.

sp> Suggesting one response or another is simply
sp> expressing an opinion of the designers' competence

Yup, but not "simply". It is an expression of the thirty year's history of software engineering. It is an expression of the difficulty of understanding the informational milieu, both external and internal, of software. It is an expression of the historical fact that we consistently fail to predict all the relevant factors, and are thus forced to learn from experience. It is not a claim that even the most brilliant team of individuals could do better.

sp> rather than any realistic assessment of the risks of SDI.

History certainly suggests this is a realistic assessment -- although I admit that a complete assessment of the risks requires greater length than our Dear Moderator would be willing to allow, or than many mailers could stand.

[[DM = Dear Moderator]]
[DM> THERE IS NO SUCH THING AS A COMPLETE ASSESSMENT OF RISKS. PGN]

sp> Given that neither the design nor the
sp> designers are determined yet, this is a silly exercise.

Nope. It is called looking to history for guidance.

Risks of raining computer printout

Alan Wexelblat <wex@mcc.com>
Thu, 30 Oct 86 10:40:59 CST

This is an old one from my viewpoint. At Penn, there is an event called Primal Scream Night, which occurs on the Sunday night before the first Monday of finals. Students are encouraged to let off steam by yelling and tossing paper (an occasional notebook or Econ text has been known to fly).

Anyway, in anticipation of this event, students raided the waste bins at the computer center, acquiring many reams of junked output as well as boxes full of punch-card holes. The next morning, we went down to breakfast early and to relieve the boredom we started reading some of the fanfolded output:

"Gee, here's a list of all the CSE110 accounts" [> 300 names]
"And here are the randomly-generated passwords."
"I'll bet nobody's bothered to change their passwords"

Sure enough, we found dozens of "available" accounts. It seems that the monthly accounting run had been done that Sunday and the output had been appropriated before the janitorial service had come around to dispose of it.

Several RISKS violations can be seen here:

- leaving a paper trail of information that should be secure
- not disposing of said paper in a secure manner
- not forcing users to change their passwords (ever)

Still, it was lots of fun to see the look on the comp center director's face when we handed him the printout and he realized what it was.

Alan Wexelblat
ARPA: WEX@MCC.ARPA or WEX@MCC.COM
UUCP: {seismo, harvard, gatech, pyramid, &c.}!ut-sally!im4u!milano!wex

---

Risks of raining computer printout

Martin Ewing <mse%Phobos.Caltech.Edu@DEImos.Caltech.Edu>
Thu, 30 Oct 86 09:43:26 PST

How many thousand sheets per printout dropped? Indeed, this seems like a brutal risk if the sheets aren't burst and/or shredded first.

Risks of raining computer printout

Peter G. Neumann <Neumann@CSL.SRI.COM>
Thu 30 Oct 86 10:37:48-PST

I might have noted in RISKS-3.90 that I once littered New York's Central Park West with TWO MILES of printout during Charlotte Moorman's Avante Garde Festival in 1967 -- a two-mile long continuous computer-printed human-composed visual poem. My poet friend Emmett Williams and I did a bunch of such computer-aided visual poetry in the late 60's. (That year Charlotte led the parade playing her 'cello suspended from helium balloons.) I had rigged up my station wagon to have Bell Labs' computer music emanating from roof-mounted speakers and computer-generated murals of Ken Knowlton on the sides of the car, with Emmett nursing the printout out of the back window to cover the middle stripe of CPW. It was wonderful to see kids rushing out between the moving vehicles, tearing off some of the printout for souvenirs! The computer RISK lay in the fact that our then-developing Multics system bellied up for a day or so when -- having used Ken Thompson's QED to context-edit an incredibly lovely 7-language interwoven visual pun -- I was ready to prepare the printout. A simpler substitute had to be used, produced by an alternative means. (The show must go on.) PGN