Systems at Cornell

Andrew Myers
March 4, 2005
Cornell Computer Science: A Unique Environment

• Recognized leader in systems research
• Broad coverage of the systems area
• Vibrant group with strong students
• Collaborative environment
• Solving fundamental problems
• Building real implementations
• Impact
What is Systems?

- A broad area...
  - Architecture
  - Databases
  - Data mining
  - Security
  - Distributed computing
  - Peer-to-peer systems
  - Fault tolerance
  - High-performance computing
  - Mobile systems
  - Networking
  - Operating systems
  - Language design and implementation
  - Program analysis
  - Compiler optimization
Researchers in Systems

– David Albonesi
– Ken Birman
– Martin Burtscher
– Alan Demers
– Paul Francis
– Johannes Gehrke
– Zygmunt Haas
– Rajit Manohar
– José Martínez
– Sally McKee
– Andrew Myers
– Keshav Pingali
– Robbert van Renesse
– Radu Rugina
– Fred B. Schneider
– Jayavel
– Emin Gün Sirer
– Steve Wicker
Systems: The Big Picture (not to scale)

Architecture

Networks

Operating Systems

Security

Languages and Compilers

Databases
Security at Cornell

• Security is an important, growing concern
• A cross-cutting concern:
  – Operating system security
  – Network security
  – Language-based security
• A fundamental problem
• Perfect for Cornell: a leader in computer security
Security Research Projects

Fred Schneider

• Editor, *Trust in Cyberspace* NRC report
• Co-Chair, Microsoft Trustworthy Computing Initiative
  Academic Advisory Board
• Integrating security and fault tolerance
  – How to build trustworthy systems from untrustworthy components?
  – COCA online certificate authority
• Inlined Reference Monitors
  – Automatically instrumenting code with security enforcement

Gün Sirer

• CorSSO: Cornell Single Sign-On
  – No centralized authenticator: an open marketplace
Security Research Projects

Dexter Kozen
• Efficient Code Certification (ECC)
• Verifying firmware

Andrew Myers
• Program analysis for information security
• Making distributed systems secure and reliable by construction
Trustworthy systems

Amazon.com Privacy Notice:

We reveal only the last five digits of your credit card numbers when confirming an order. Of course, we transmit the entire credit card number to the appropriate credit card company during order processing.

Promotional Offers: Sometimes we send offers to selected groups of Amazon.com customers on behalf of other businesses. When we do this, we do not give that business your name and address.

Protection of Amazon.com and Others: We release account and other personal information when we believe release is appropriate to comply with the law; enforce or apply our Conditions of Use and other agreements; or protect the rights, property, or safety of Amazon.com, our users, or others.

...Promises, promises.
How does Amazon know they meet legal obligations?
Secure distributed systems?

• How to build?
  – encapsulation, access control lists,
    distributed protocols, encryption, signing,…

• How to validate?

• Our goal: systems secure by construction
  – Programs annotated with explicit security policies
  – Compiler/static checker checks, transforms programs to satisfy policies
Distributed Battleship

- Two-player game in which each player tries to sink other’s ships

- General problem for multiplayer games/simulations: hard to prevent cheating
  - Distrust $\Rightarrow$ Multiplayer code must change.

- Almost any distributed system has distrust
  - Online shopping, financial systems, trading, B2B, email, user profile management, military information systems with dynamic coalitions

- Idea: transform code to run securely on untrusted hosts
Automatic partitioning and replication
(SOSP, IEEE S&P)

Describes the trust relationships between principals and hosts.

Source Code (Jif)

Verifies that the program obeys the security policies. A subprogram is split into multiple pieces and sent to multiple hosts, so that policies are obeyed.

Describes the computation and the principals' security policies. A subprogram may be replicated on multiple hosts.

Partitions the data and computation among hosts, so that policies are obeyed.

Trust config

Compiler Splitter

subprograms

Host 1

Host 2

Host 3

Host 4

Host 5

network protocol
Battleship example

- A’s board is confidential to A but must be trusted by both A and B
- B’s board is symmetrical
Using replication

• **Idea 1**: replicate both boards onto both hosts so both principals trust the data.

• **Problem**: host B now has A’s confidential data.

• **Idea 2**: host B stores a one-way hash of cells
  • Cleartext cells checked against hashed cells to provide assurance data is trusted by both A & B.
  • Compiler *automatically* generates this solution!
And much, much more…

- Architecture
- Operating Systems and Networks
- Databases and Digital Libraries
- Languages and Compilers
Architecture

- **Martin Burtscher: High-performance multiprocessors**
  - Fast, energy-efficient hardware architectures
  - High-performance program trace compression
- **Rajit Manohar: Asynchronous VLSI**
  - Async. FPGAs as reconfigurable dataflow machines
  - Nanowatt computations for sensor networks
- **José Martínez: Thread-level parallelism**
  - Checkpointed processor architectures, thread-level speculation, reconfigurable multiprocessors
- **Sally McKee:**
  - System-wide hardware/software performance monitoring and adaptation