Biology is the Science of Security

Stephanie Forrest
UNM and Santa Fe Institute
March, 2008

THE UNIVERSITY of NEW MEXICO

What can we learn from other fields?

- Experimental design
  - How to conduct experiments and analyze results
- Quantitative methods
  - PCA, ICA, nested models, species-abundance curves, phylogenetic tree reconstruction, power law analysis. How to evaluate results based on unfamiliar methods? Do the theorems provide insight?
- Architecture, mechanisms, and principles of other complex systems
  - Study solutions that have been developed in other systems to problems that are similar to those we want to solve

Experiments
seems obvious but ...

- Conducting repeatable experiments
  - Articulate a clear hypothesis and design the simplest possible experiment. Allows others to confirm results and test variations
  - Public domain prototypes and data sets (overfitting issue)
- Careful comparisons and repeatability are surprisingly difficult
  - Complex environments
  - Results often depend heavily on data inputs
  - Metrics that emphasize breadth of coverage and corner cases

THE UNIVERSITY of NEW MEXICO

Principles of biological computation

- Traditional approach to CS:
  - Decomposability and modularity
  - Explicit management of exceptions and interactions
  - Efficiency, correctness, and optimality
- Lessons from biology:
  - Survivability and evolvability
  - Autonomy
  - Robustness, disposable components
  - Adaptation and self repair
  - Diversity
  - The cost of getting big

THE UNIVERSITY of NEW MEXICO

Biological defense mechanisms
Applied to computation

- Immunology:
  - Protect an individual (single host or a network) against network epidemics and other forms of attack.
  - Antivirus programs, intrusion-detection systems
  - Sana Security *Primary Response*
- Autonomic responses, e.g., homeostasis:
  - Tightly coupled low-level detection/response phases.
  - pH and network (virus) throttling.
  - *HP's Virus Throttle*

Biological defense mechanisms Applied to computation cont.

- Diversity:
  - Genetic diversity leads to population-level robustness.
  - Disrupt software monoculture using randomization and/or evolution.
  - *Microsoft Vista Address Space Randomization*
- Epidemiology:
  - Network-based control of viruses/worms.
  - Focus on network topology (the epidemic threshold).
  - Survivability and attack resistance (PGBGP---work in progress)

Other biological defense mechanisms
Still to be tapped

- The innate immune system
- Ecological interactions and evolutionary biology
  - Malware ecology: Malware interactions, indicator species, etc.
  - Automated bug repair using evolutionary methods
  - Optimal levels of defense in depth
- Intracellular defenses and repair mechanisms
  - RNA_i
  - Restriction enzymes

Overarching themes

- What level of abstraction is appropriate?
  - Negative selection mechanism vs.
  - Automated diversity
- What makes a computation *biological or biologically inspired?*
  - Architecture, mechanism, functionality
- Biological principles are being discovered in bits and pieces
  - Need a unified framework

THE UNIVERSITY *of* NEW MEXICO

Science envy?

- We may have made more progress than we realize
  - Forcing attack vectors to evolve
- Why should we expect to *solve the problem* so that we never need to touch it again?
  - Biomedicine doesn’t, economics doesn’t
  - No simple quantitative metrics for “health”; Indicators rather than metrics?
- Suggestion: “Accumulate knowledge in a systematic fashion”
- It’s not only about quantitative prediction (building intuitions, existence proofs, critical regions)

Engineering practices
based on principles of biology

- Why do we need them?
  - Evolution of the software ecosystem (software rot, malware)
  - Dynamic, mobile, complex, and hostile environments
  - Moore's Law won't rescue us
- Hallmarks
  - Simple and generic
  - Computationally and memory efficient
  - Automatically self-tuning, distributable, diverse, and autonomous

THE UNIVERSITY of NEW MEXICO