

# On the Future of Cybersecurity\*

Jonathan M. Smith  
Jms@cis.upenn.edu  
UIUC, 2/20/08

\* Warning: This talk is meant to provoke!



# Our model for “secure” is flawed

- Idea: software can be *proven* correct\*
  - **Security** is a correctness property
  - Therefore, we can verify software and obtain secure systems!
- This thinking is based on the foundations of CS in logic and formal languages
- Where has it gotten us (in >40 years)?

\*Not everyone believed this. For a very interesting read, see De Millo, Lipton and Perlis, “Social Processes and Proofs of Theorems and Programs”, CACM May 1979

# So what's wrong?

- Secure/NOT Secure model has axiomatic assumptions, used as scaffolding for proofs
  - Proofs are *independent* of the environment
  - But how *could* they be?
- Security failures when assumptions *violated*: interactions w/new environment (including new compositions with other systems)
  - Assumptions are always violated - very secure systems are standalone (“air gapped”)!
  - Standard cracker practice: violate!

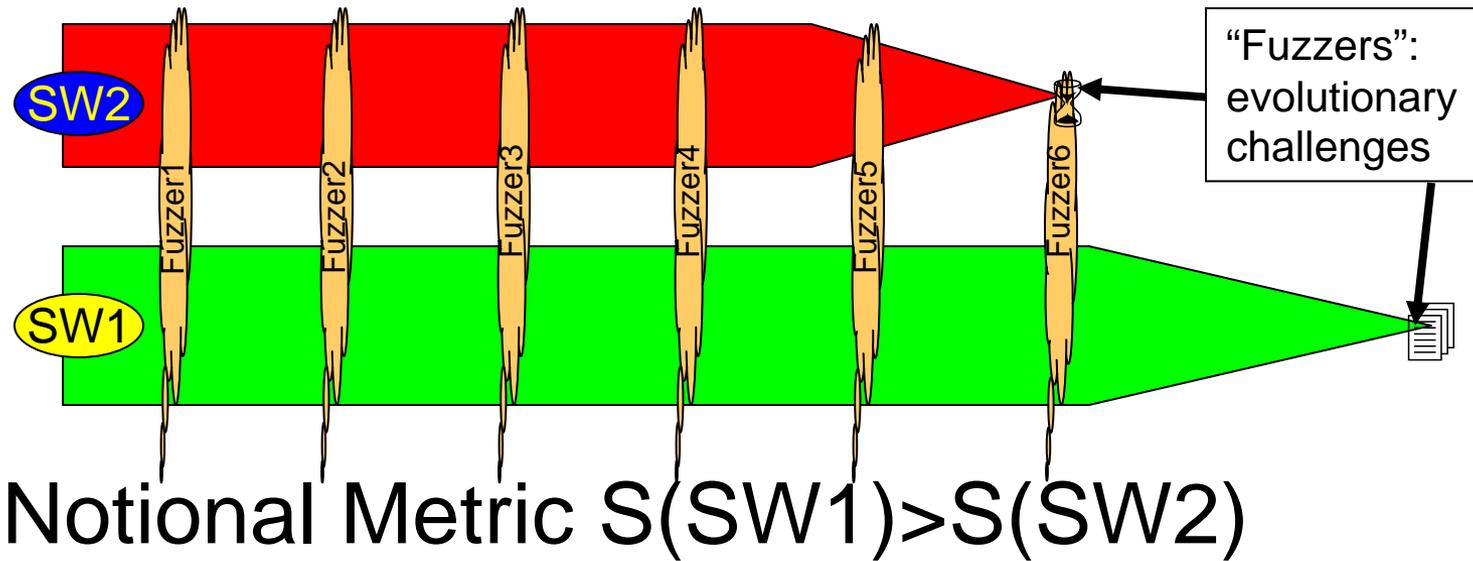
# Continuous Security

- The environment is continuous and evolving
  - Software has to live in a cyber “ecosystem”
- Can we develop a model that models:
  - New compositions (e.g., SOA)?
  - Evolving environments? (e.g., cellphones)
  - Evolving threats? (e.g., botnets)
  - Response to a range of unknown threats?

# Software Evolution?

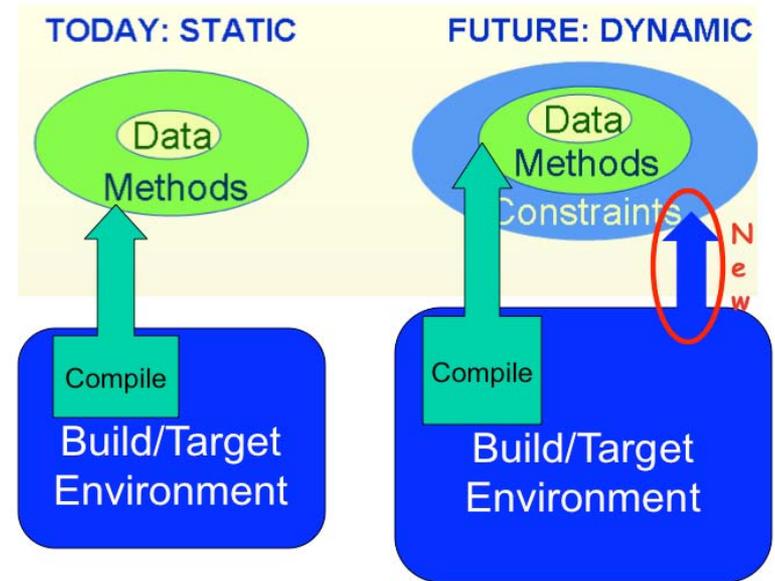
- T. S. Ray, "Software Evolution." *Systems, Control and Information* 40(8): 337-343, 1996.
  - Tierra system
- McKinley, *et al.*, "Harnessing Digital Evolution", *IEEE Computer*, Jan. 2008.
  - Avida system
- Great idea – will need to overcome farmed versus natural environment issues...

# Can we measure evolutionary fitness for software?



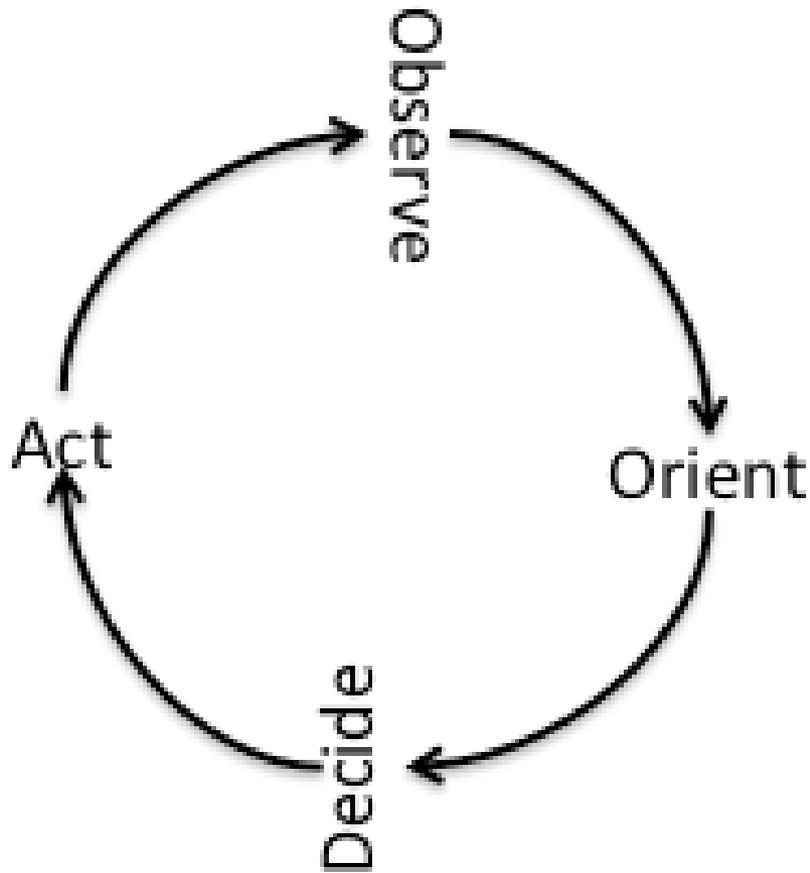
# Maybe Software Evolution versus Creation?

- Preserve good components
- Composition / inheritance
- Embrace complexity\* - the environment is complex!
- Save *everything* that worked in the past
- Cognitive: sense, compute, actuate



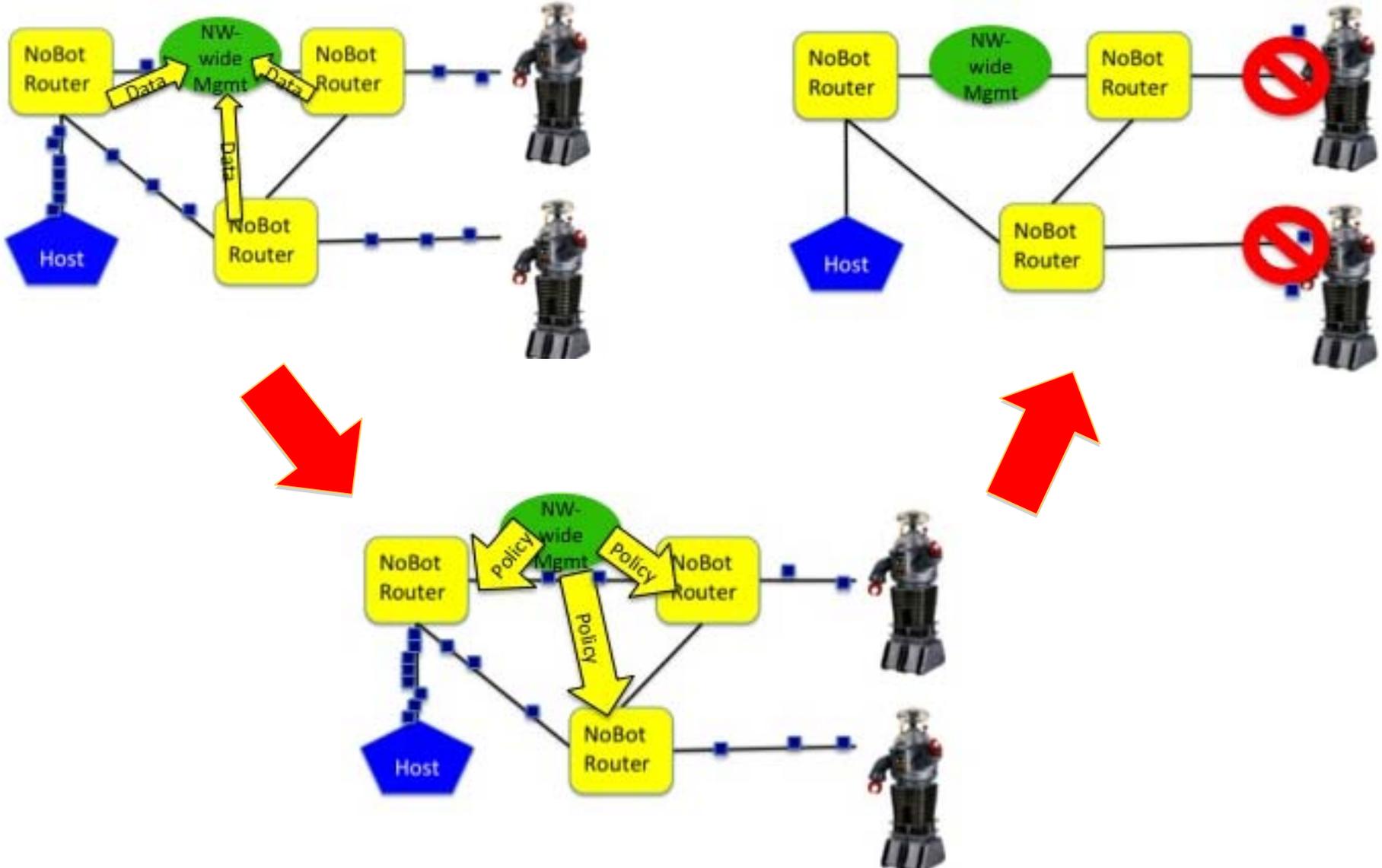
\* The best essay I have read on scale is J.B.S. Haldane's "On being the right size". All engineers should read it!

# John Boyd's OODA Loop



- Faster cycles than adversary: wins
- Technologies should therefore focus on accelerating OODA loop cycles

# Networks opposing Botnets



# Conclusion

- We need to rethink cybersecurity strategically
  - Goal: Measurable successes
- Craftsmanlike software engineering can't scale
  - Need new ways for software production
- Evolutionary / ecosystem models make sense
  - Apply Boyd's OODA loop
  - The network is an important part of the solution