SAND2009-4192C

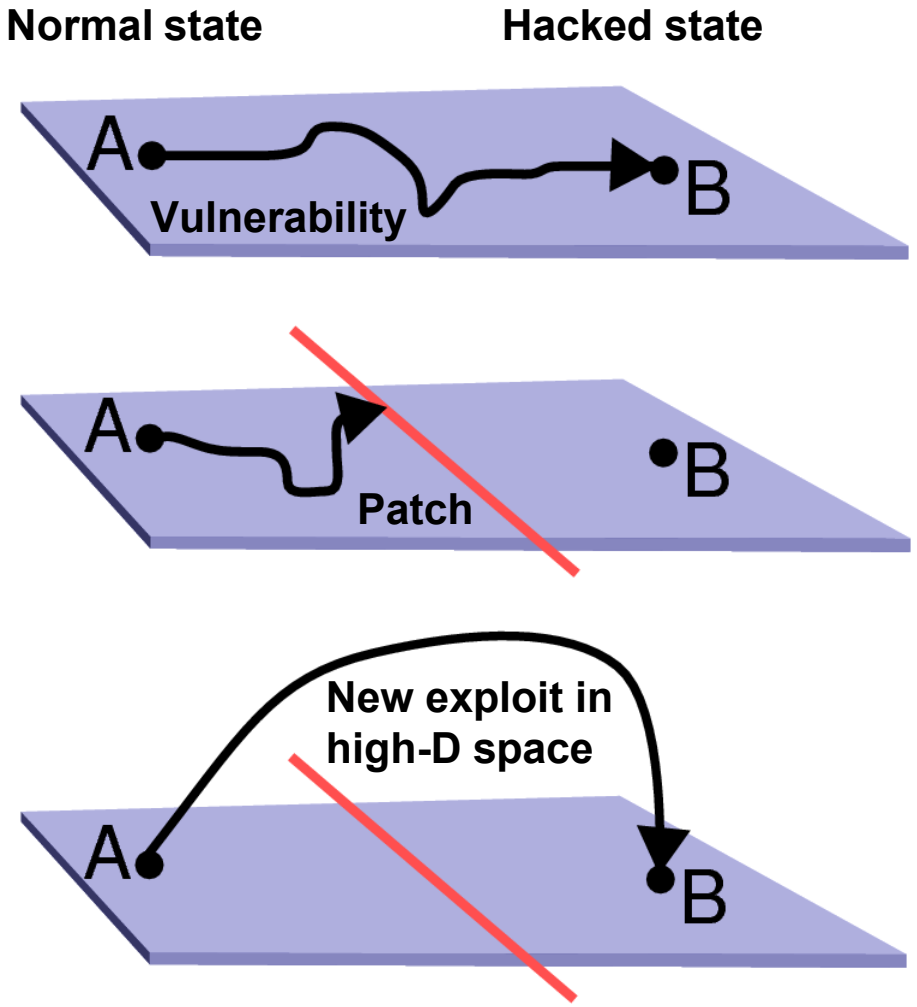
Office of Science Mathematical and Computer Science Approaches to Cybersecurity

Rob Armstrong
Sandia National Laboratories
Livermore, CA

Sandia National Laboratories

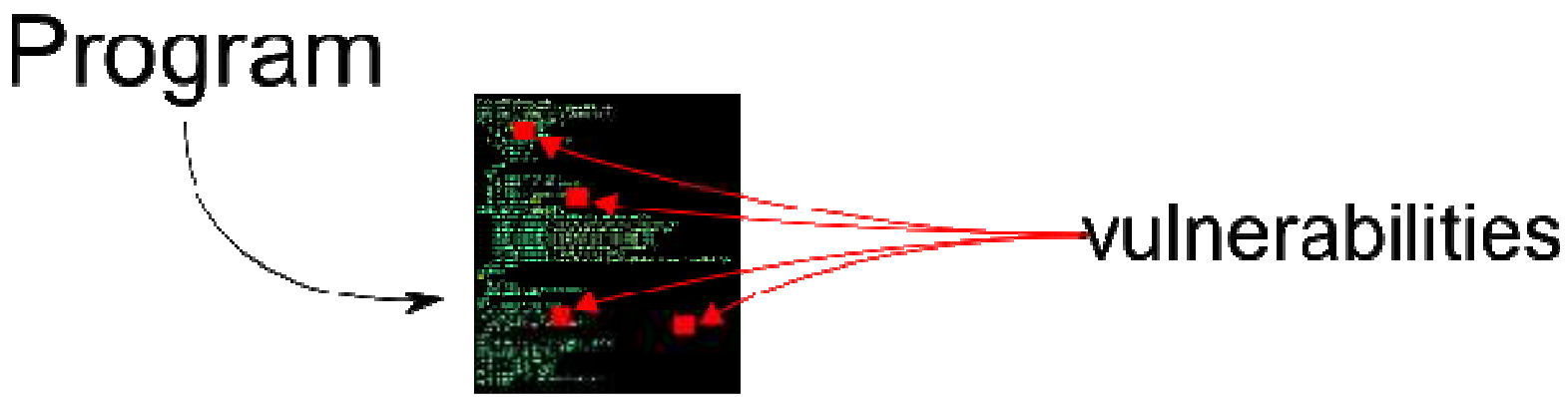
Combinatorial explosion generates fundamental asymmetry of cyber defense

- Vast state space defies enumeration, even for desktop computers; new attacks every day
- Turing completeness makes behavior effectively unpredictable
- Numerous hidden attack vectors give attackers asymmetric advantage
- Current cyber defenses amount to one “Maginot Line” after another



Securing an arbitrary code is not just hard; it's impossible

- **Restated: Generic code has vulnerabilities that are unprovable and unknowable**
  - *Not* statistical, even in principle
  - Turing completeness demands that a generic code is undecidable



- **So now what?**

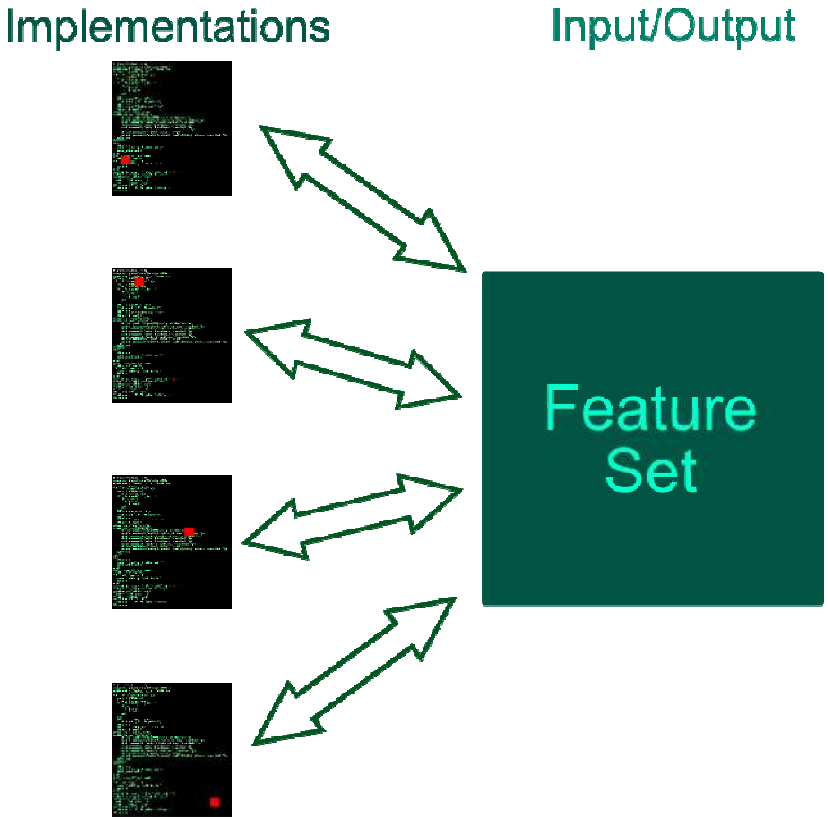
Complexity makes cyber threats *asymmetric*

Bad Guy needs to find one

You have to find them all

- **Developer, user, *and attacker* all don't know where the vulnerabilities are (*undecidable*)**
- **Asymmetry: One vulnerability compromises the whole code**
  - Developer has to find all of them (impossible in general)
- **No one can guarantee "this code is clean" or even quantify improvement**

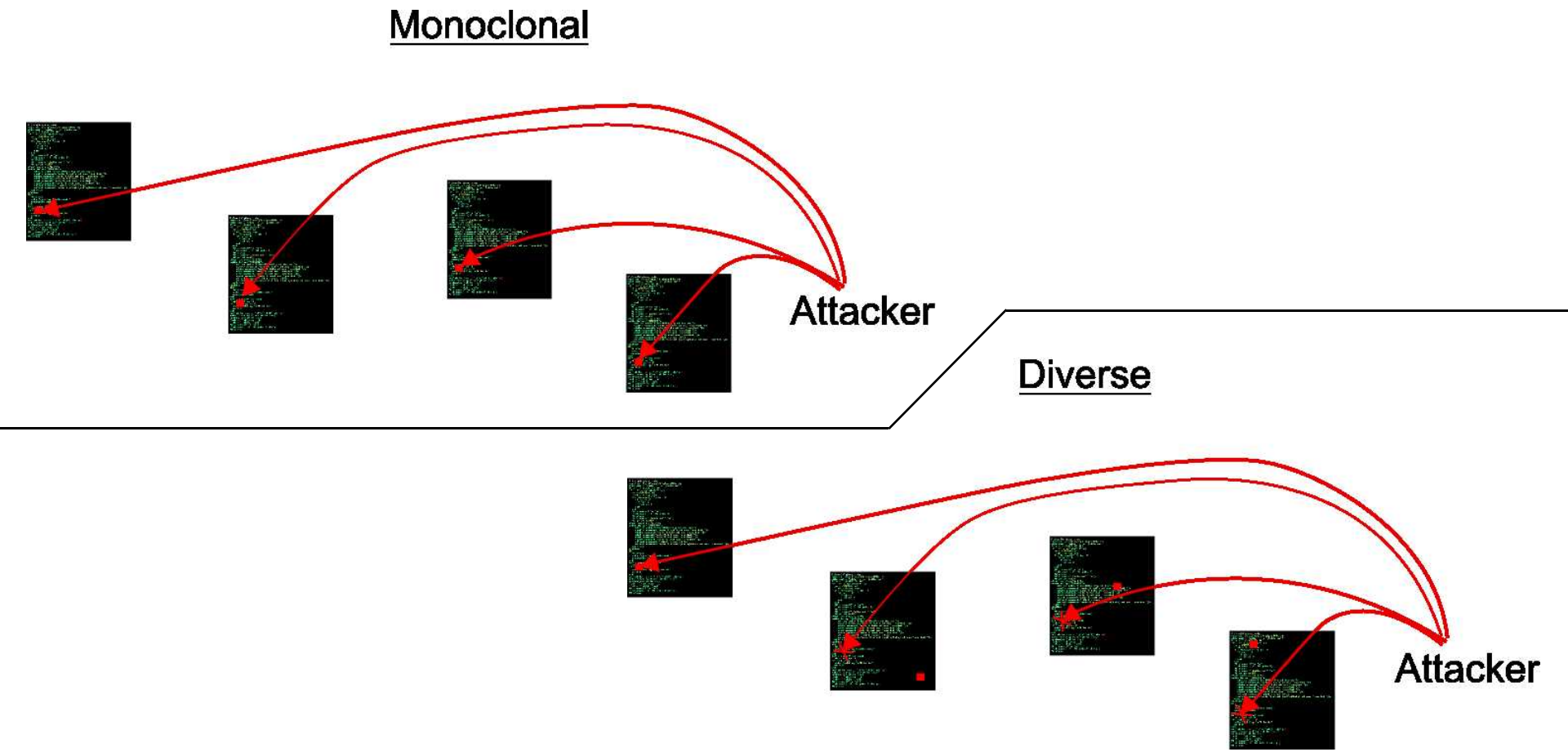
Observation #1: A program's feature set has many implementations



- **Feature set is defined by a test suite**
- **Test suite verifies that an implementation conforms to desired functionality**
- **Test suite is a sample; cannot realistically cover all possible input/outputs**
- **Vulnerabilities arise from untested input/outputs**
- **Any feature set has infinitely many implementations**
  - Finite large number if size is bounded

Observation #2: Ensemble of instances permits the formulation of statistics

- **Assume: Multiple implementations randomize security holes**
- **Ensemble of multiple-version, "randomized" undecidable codes allows formation of security *improvement* statistics**

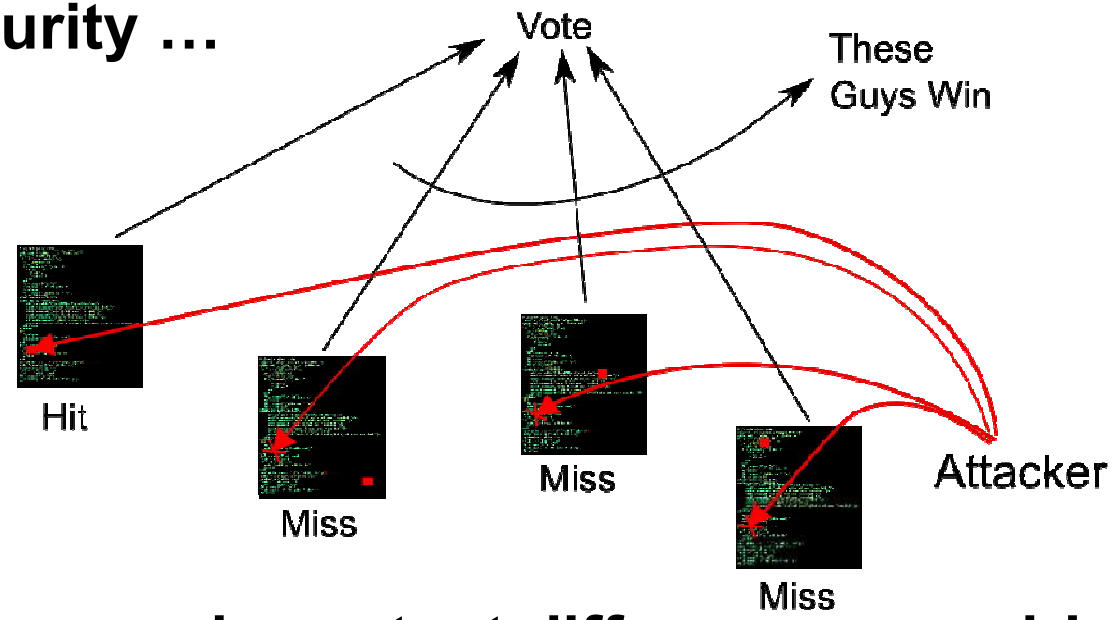


High-reliability systems can be constructed from "*N*-version software"

- **Space Shuttle: 4 computers, identical software, different hardware, same design**
  - Focus is on hardware faults
- **Similarly, software redundancy used mostly for control systems up to now**
  - *N*-version software: Multiple versions implemented to the same feature set by different developers
- **Models of *N*-version software view the control system as a stochastic process that walks the code graph of the software**
  - Control system takes the place of a "fuzzer"

Similarly, *N*-version software can quantifiably improve cybersecurity

- **Clear generalization of *N*-version reliability to cybersecurity …**

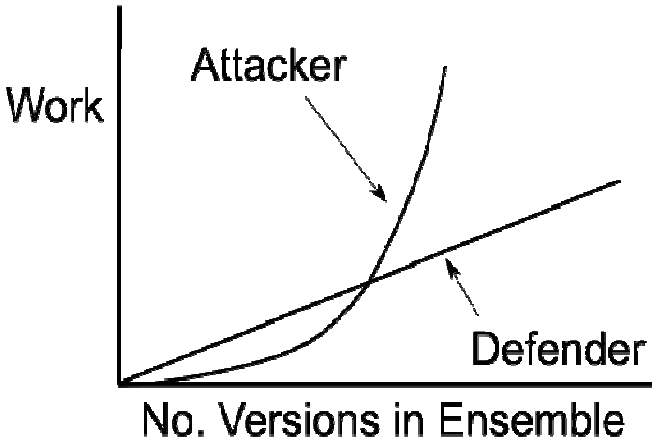


- **… but there are important differences requiring enabling technology**
  - Compromised versions must be removed and replaced
  - Hand-made new versions are time-consuming and expensive
    - **May repeat previous mistakes**

Sandia National Laboratories

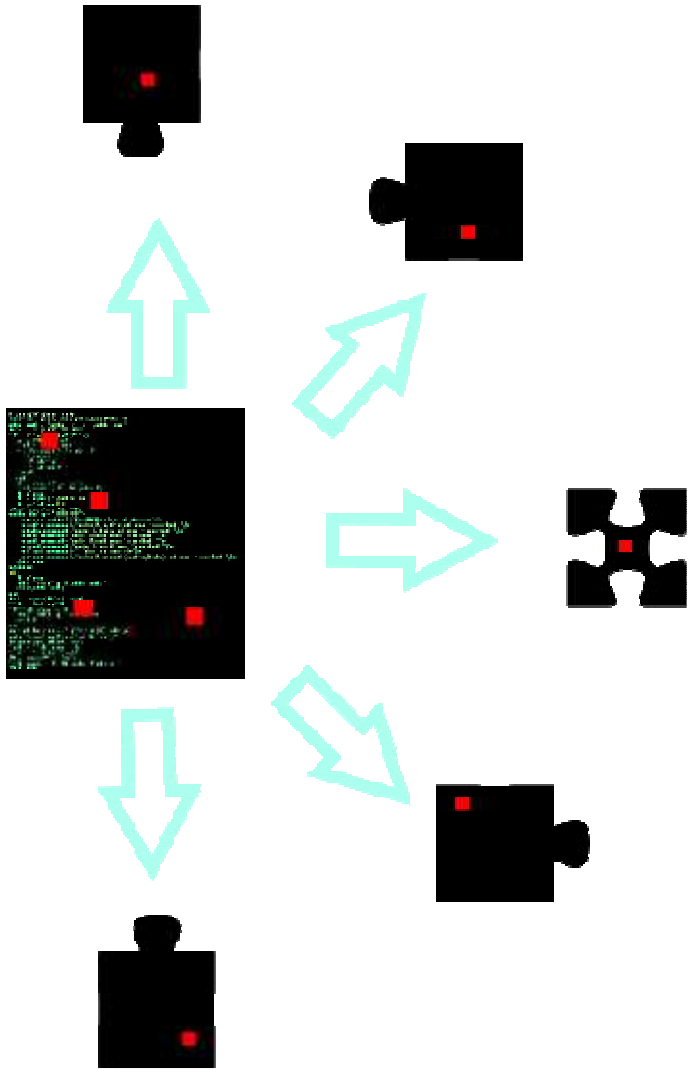
Simple Statistics from an Ensemble of Undecidable Programs

- **On a specific feature set (*F*) there is a probability (P_v) that a particular sample from the set of implementations of *F* will be susceptible to vulnerability (*v*). For a voting ensemble of size *n*:**
  - The probability of success (or the "work") for the attacker is: $(P_v)^{-n/2}$.
  - The work for defender is the cost of producing n implementations: ✪ n

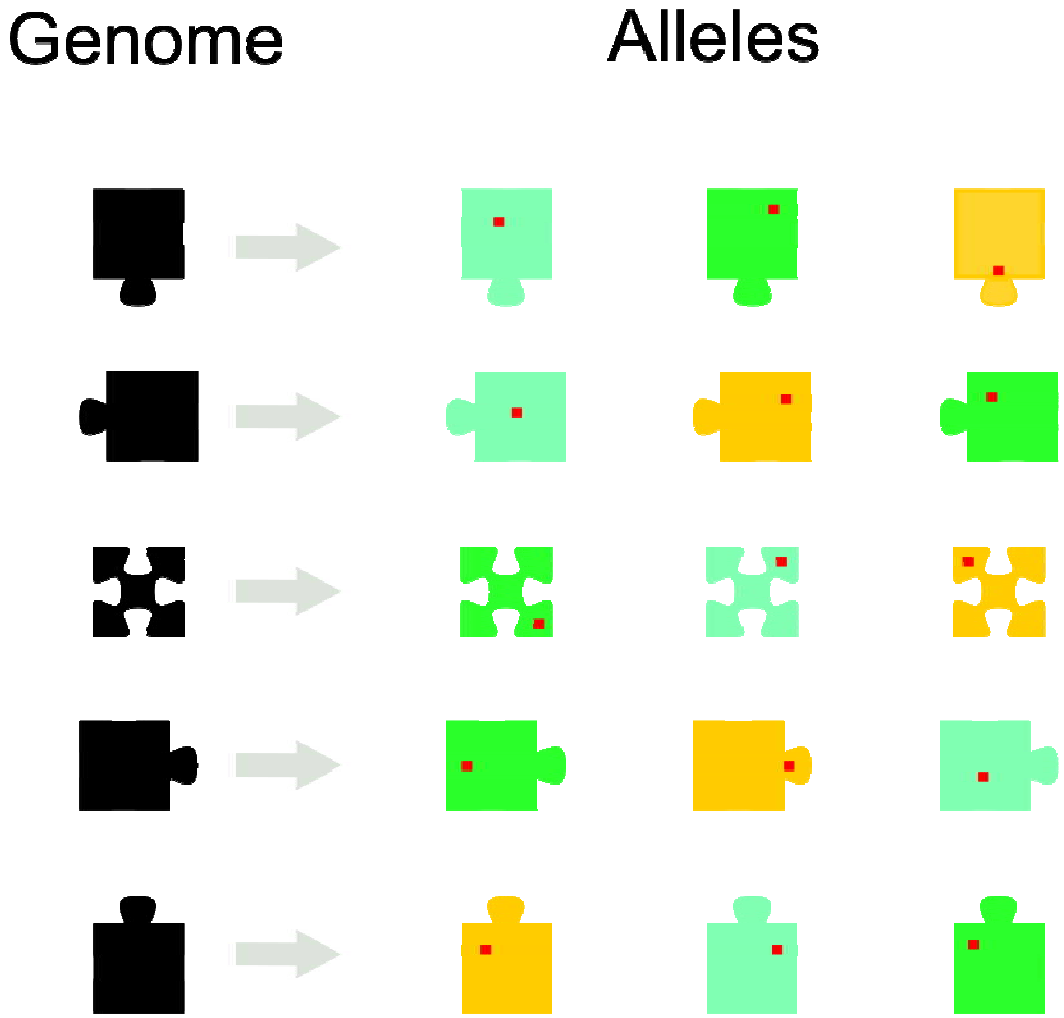


Sandia National Laboratories

Break a software implementation into components



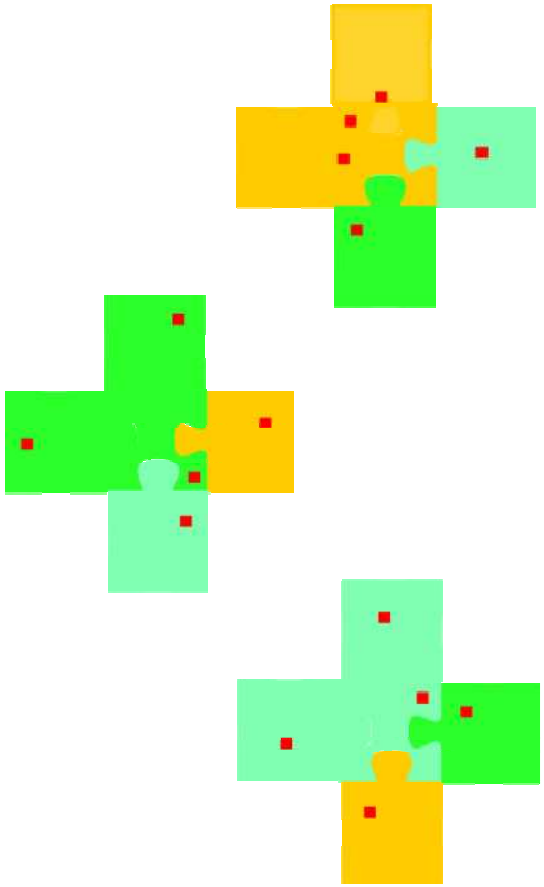
Living systems adapt to cope with unknowable attacks



- A component type is similar to a gene; component implementations are similar to alleles of a gene

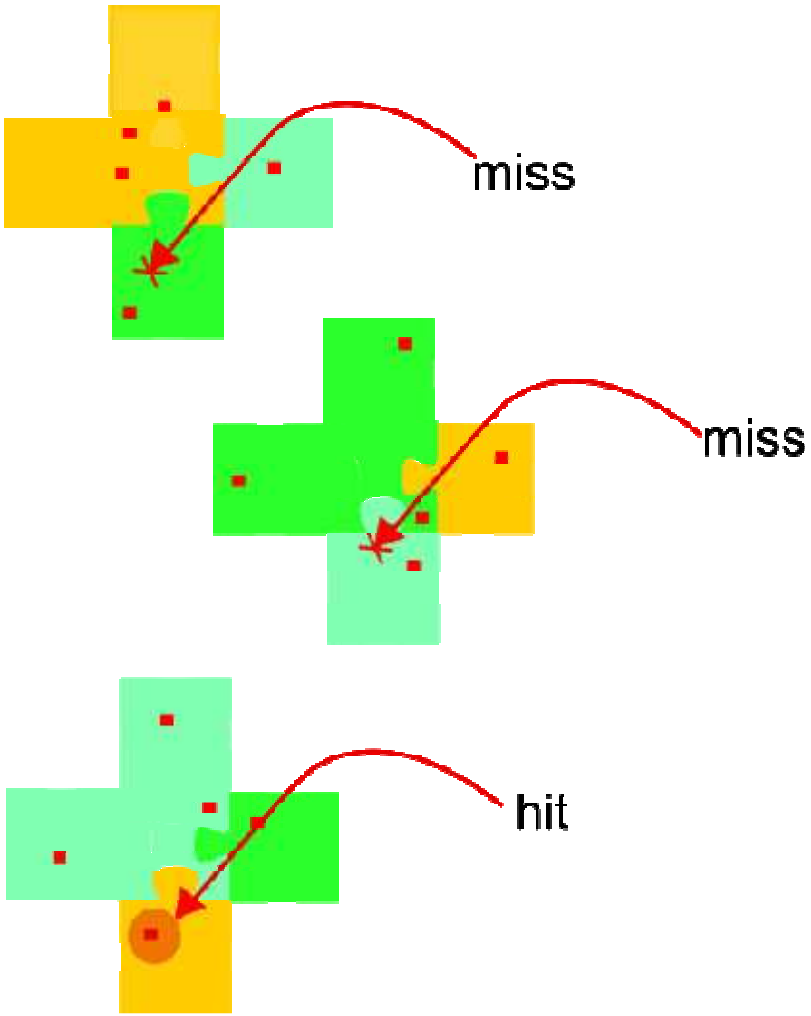
Sandia National Laboratories

Reassemble alleles into individuals



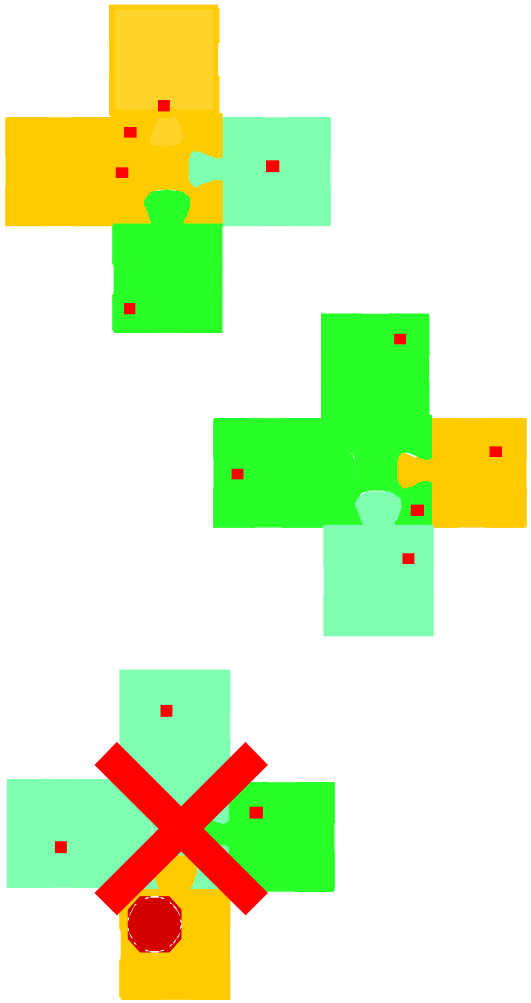
- Different alleles can be assembled into new individuals that have “randomized” security holes
- New individuals are differently vulnerable and potentially adaptive

Compare responses from individuals



- Now different individuals will produce the same feature set but react differently to attacks

Evolve new and more robust individuals

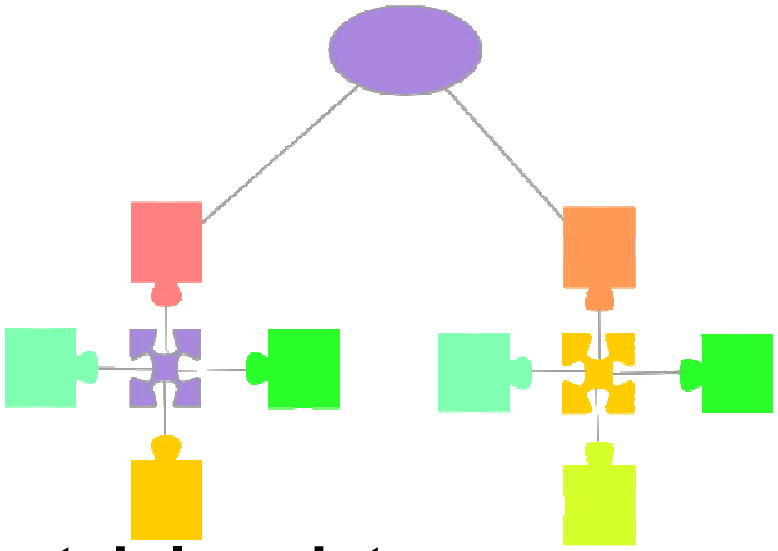


- Eliminate the one with the differentiated response

Sandia National Laboratories

Genetic approach is a special case of fault-tolerant system design

- **Previous genetic approach is a contrived example of a network of entities robust to attack …**

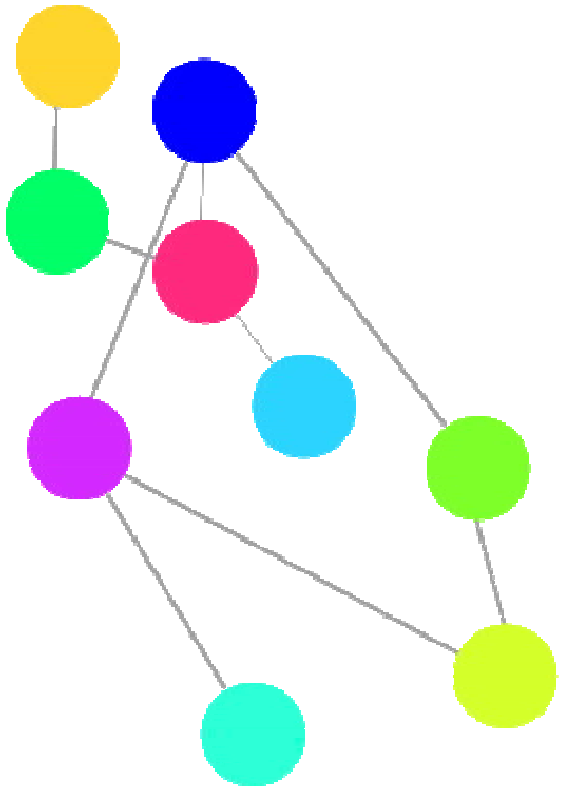


- **… others certainly exist**
  - Seek more efficient examples that do not require total replication of every attackable system
  - Seek an arrangement of entities that has no single point of failure

Research needs: Quantify and automate "program randomization"

- **Exploit existing compiler code-rewriting techniques**
  - Stack randomization, semantic invariant transformations, etc.
  - Obfuscation techniques
- **Quantify, or at least *qualify*, sufficient randomness to expect that vulnerabilities do not repeat**
  - Use "fuzzers," emulation and automated software analysis to find and compare *some* implementation-induced vulnerabilities
  - Use data to model the prevalence and rate of exposing new vulnerabilities
- **Automate the process of finding new versions**
  - Genetic programming techniques hold promise
    - **We are starting from a known implementation**
    - **Seeking only diversity**

Generalization: Seek the emergent property of robustness to attack

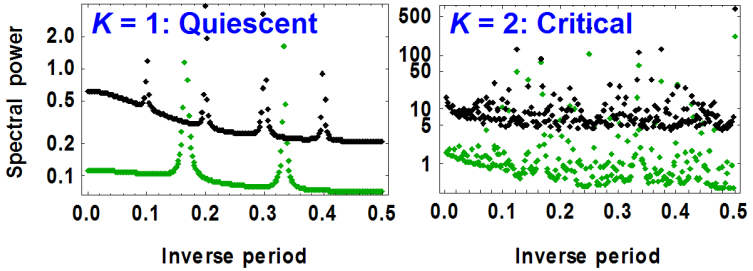


- **Seek a generalized approach similar to RAID, where both diversity and voting are incorporated in-depth**
  - Eliminate the need for complete replication of diverse programs
  - Eliminate single points of failure
- **Seek a network of entities that are collectively robust to attacks "in the cloud"**
- **Entities can generalized to**
  - Programs
  - Hardware
  - Networks
  - Systems of these systems
  - And so on …

Sandia National Laboratories

Complexity science offers new methods for cyber modeling and design

- **Many real-world complex systems (social, financial) exhibit "emergent" self-organization and robustness**
- **Complexity science provides "top-down" insight relating system structure to emergent behavior**
- **Idealized complexity models (cellular automata, Boolean networks) give foundational results**
  - Emergent behavior can be captured using coarse-grained models

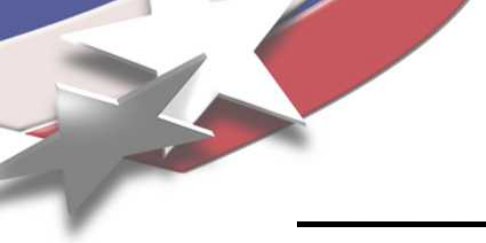


- **Entity-based models treat general complex systems**
- **"Entity-oriented programming" would use a robust "regulatory" network to maintain desired functionality**

Sandia National Laboratories

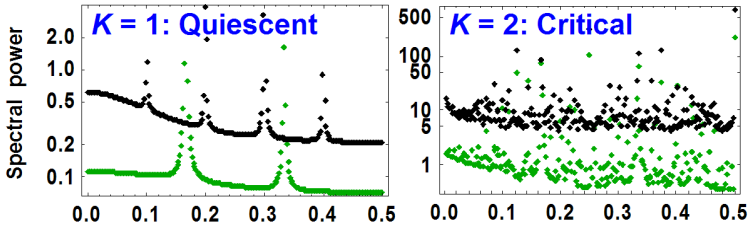
Conclusions

- **Cybersecurity can benefit from the "familiar" math and science...**
  - Graph Theory, Machine learning, Agent-Based system models
- **... but also the "unfamiliar" math and science**
  - Understanding and modeling the unknowable
    - **Deriving articulations (like ensembles) of unknowables in order to form meaningful statistics**
  - Use complexity science to inform the design of complex programs, computer systems, and networks



Complexity science offers new methods for cyber modeling and design

- **Many real-world complex systems (social, financial) exhibit "emergent" self-organization and robustness**
- **Complexity science provides "top-down" insight relating system structure to emergent behavior**
- **Idealized complexity models (cellular automata, Boolean networks) give foundational results**
  - Emergent behavior can be captured using coarse-grained models

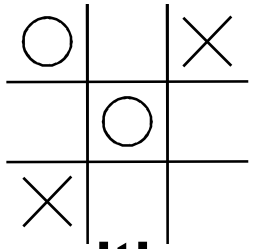


- **Entity-based models treat general complex systems**
- **"Entity-oriented programming" would use a robust "regulatory" network to maintain desired functionality**

Sandia National Laboratories

Test problem: Simple games like tic-tac-toe can model cyber vulnerabilities

- **Tic-tac-toe is a microcosm of programming challenges**
  - Move-by-move play imitates software transaction protocols
  - Perfect player is never beaten (= never hacked)
- **255,168 possible games: Nontrivially complex but tractable by brute force when needed**



- **More interesting than perfect programs are those with useful but flawed heuristic strategies**
- **To model real software, the "feature set" is specified by good moves on a *subset* of possible boards**
  - Program complexity is constrained to deter "memorization"
- **Key question: Can genetic variants, adapted to a limited test suite, play tic-tac-toe well (robustly) as a team?**

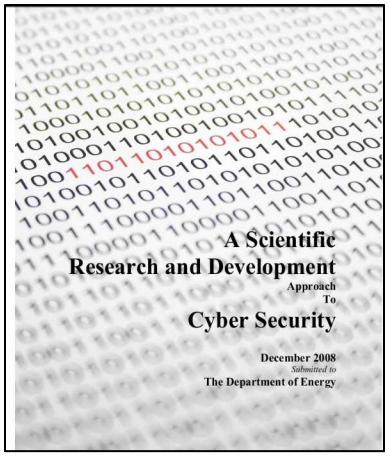
Boneyard

Sandia National Laboratories

Science Based Cybersecurity

- **DOE Office of Science seeks to establish a program in cybersecurity**
  - Directed toward DOE special needs in cybersecurity
  - Using DOE special talents
- **Report resulting from a series of workshops**
- **Concentrating on three areas:**
  - Math (Graph theory, pattern recognition, complex systems)
  - Platforms (Internet Simulation/Emulation)
  - Data ("Self" protecting data)

http://www.sc.doe.gov/ascr/ProgramDocuments/Docs/CyberSecurityScienceDec2008.pdf



Sandia National Laboratories

Graph Theory and Complex Systems Modeling

- **Network "tomography": Deduce global network structure based on limited observations**
  - From network traffic recorded at points in a network
    - **Incomplete information**
  - And regularization conditions to impose uniqueness
    - **Drawn from an understanding of what the network "should" be.**
    - **Ability to generate synthetic networks that have the observed properties of real networks.**
- **Agent-based models of network dynamics**
  - Emergent infection of a complex network under attack
  - Understand impact of topology, protocols, and configuration on attacks.