SANDIA REPORT

SAND2009-2007
Unlimited Release
Printed March 2009

Complexity Science Challenges in Cybersecurity

Robert C. Armstrong, Jackson R. Mayo, Frank Siebenlist

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of Energy's
National Nuclear Security Administration under Contract DE-AC04-94AL85000.

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online ordering: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2009-2007
Unlimited Release
Printed March 2009

Complexity Science Challenges in Cybersecurity

Robert C. Armstrong
Scalable Computing R&D

Jackson R. Mayo
Visualization & Scientific Computing

Sandia National Laboratories, P.O. Box 969, Livermore, CA 94551

Frank Siebenlist, Mathematics & Computer Science
Argonne National Laboratory, 9700 S. Cass Avenue, Argonne, IL 60439

Abstract

Computers and the Internet are indispensable to our modern society, but by the standards of critical infrastructure, they are notably unreliable. Existing analysis and design approaches have failed to curb the frequency and scope of malicious cyber exploits. A new approach based on complexity science holds promise for addressing the underlying causes of the cybersecurity problem. The application of complexity science to cybersecurity presents key research challenges in the areas of network dynamics, fault tolerance, and large-scale modeling and simulation. We believe that the cybersecurity problem is urgent enough, the limits of traditional reductive analysis are clear enough, and the possible benefits of reducing cyber exploits are great enough, that the further development of cybersecurity-targeted complexity-science tools is a major research need.

1 Introduction

Computers and networks of computers are indispensable to our everyday work because they are so powerful as information processing and communication tools. Partly as a result of this immense capability to do the things we want, computers will also do many things that we do not want. In fact, computers are increasingly doing the bidding of attackers, to the detriment of their owners.

Complexity is not an incidental feature of cyber systems but an inextricable necessity due to the complex things they are required to do. Complexity allows serious vulnerabilities to exist in apparently benign code, and allows those who would exploit them to attack invisibly. To understand the cybersecurity problem, we must understand cyber complexity and model it with predictive simulation. To solve the cybersecurity problem, we must figure out ways to live with the inherent unpredictability of modern cyber systems.

In the face of the essential unpredictability of software and the ability for attackers to remain unseen, there are three major approaches to mitigation, all of which will benefit from a more scientific approach:

1. **Model checking, formal methods** [12]**, and software analysis** [22] detect errors and, in the case of very simple systems, rigorously verify behavior as long as the foundational assumptions are correct. Most realistic cyber systems are too complex for rigorous verification, but can benefit from non-exhaustive analysis that will find a few of the straightforward vulnerabilities. Applications are common in supervisory control and data acquisition (SCADA) and medical devices [14], where the systems are less complex and the consequences of faults more severe.

2. **Encapsulation, sandboxing** [5]**, and virtual machines** [25] provide a way to "surround" otherwise unpredictable software, hardware, and networks with software or hardware that is more trusted. A common but often ineffective example is a network firewall. Other more effective examples are being developed [17] and this technology holds particular promise for environments unique to DOE (Section 4).

3. **Complexity science** [27] **drawing on biological** [11, 15] **and other analogues** [8] is the least exploited but possibly the most promising approach. Biological metaphors are part of the cyber lexicon: virus, worm, etc. Models of complex cyber systems and their emergent behavior are needed to understand the cybersecurity problem.

In this paper we will concentrate mainly on item 3, complexity science, and secondarily, item 2, the encapsulation of complexity. Within the complexity arena, we recommend two basic research thrusts:

1. **Confronting unpredictability in programs, machines, and networks** and its impact on the cybersecurity problem. Whether it is considered a theoretical certainty or a pragmatic heuristic, it is impossible to "formally" ensure [16] that a realistic program or operating system is

vulnerability free. Research is needed to understand how this complexity asymmetrically favors the attacker in current systems, and how it can instead be leveraged against the attacker to the defender's advantage. Theories and algorithms are needed that use complexity to increase the effort required of attackers and reduce their likelihood of success. Here existing work from the fields of fault tolerance and high-reliability systems can be helpful.

2. **Modeling the emergent behavior of programs, machines, and networks** to understand what they are doing and predict what they will do, to the extent permitted by the underlying complexity. This type of modeling is largely unfamiliar to scientific computing but enjoys wide acceptance in systems analysis (e.g., war gaming, social simulation, business operations). Agent-based and discrete-event simulation methods are commonly used in this area and have already been applied to network and computer system modeling on a small scale. Large-scale modeling and simulation relevant to Internet proportions are needed to understand how the emergent behavior of large numbers of computers, routers, and network links can either mitigate or exacerbate the cybersecurity problem. Understanding the emergent behavior of cyber systems can also enable a broader approach to reliability and fault tolerance.

2 Confronting unpredictability in programs, machines, and networks

Complexity of computers and software is an artifact of the complex things we require them to do. Their capacity for computation is inextricably connected to the fact that they are also unpredictable, or rather capable of unforeseen emergent behavior. Vulnerabilities[1] are one of those behaviors. A complex system's emergent behavior cannot be predicted from even a perfect knowledge of the constituent parts from which it is composed, a ramification of undecidability [4] and Turing Completeness. This means that a sufficiently complex system has emergent behavior that cannot be predicted ahead of "running" it, or alternatively, simulating that system with sufficient fidelity (and complexity) to see the emergent behavior [27] (Section 3). Concretely, this is why cyber systems that are composed of elements like programs, processors, and routers, each of which we presumably understand, are nonetheless constantly surprising us with the consequences of previously unknown vulnerabilities.

2.1 Origins of the cybersecurity problem in complexity

Even though most realistic applications are too complex, there are programs and systems that are simple enough to be analyzed by formal methods and other reductive tools. In those cases, boundedness properties of the code can be asserted and proved, making the behavior well-understood

[1] By "vulnerability" we mean some flaw in the system that can be exploited by a malicious actor to cause an effect not desired by the operator of the system. Note that the malicious actor could actually be a programmer or designer of the system with the intent of subverting a user.

under a wide variety of circumstances. Formal verification [12] is accomplished by automatically following all salient execution paths to understand their consequences. However, probably the vast majority of codes are of the sort that are too complex for this analysis, the number of paths soon grows beyond the capacity of even the largest machines.

In this section we consider systems that are undecidable and for which vulnerabilities can only be discovered anecdotally but not thoroughly. Because their containing programs are unanalyzable, these vulnerabilities cannot be guaranteed to be found by any means. Because the defender cannot be certain to find all of the vulnerabilities in a system, and because only one vulnerability is needed for it to be compromised, the cards are asymmetrically stacked in the attacker's favor.

2.2 Fault tolerance and high-reliability systems

A vital emergent behavior of many real-world complex systems, particularly biological ones, is robustness to disturbances. The concept of highly optimized tolerance [11] describes highly engineered systems that take on an organic structure after many design cycles promoting robustness similar to biological evolution. Usually confined to medical devices [14] and aerospace avionics [26], fault-tolerant systems use diverse redundant systems and "vote" the answers to common input, detecting faults[2] as outliers. An example is the set of identical computers on board the Space Shuttle [26]. If we recognize that a vulnerability is just a fault that is exploited, it is clear that cybersecurity too has much to gain from employing diversity [24]. To achieve robustness to such failures through redundancy, the replicated components can employ *diverse implementations* of the same functionality. In this way, exploits that trigger deviations from the intended behavior are unlikely to affect more than a small fraction of components at a time, and can be detected by voting the outputs [10, 18, 21].

Diversity need not involve complete replication of cyber systems but may be applied creatively, possibly in analogy to the way RAID encoding [2] allows any one element of a disk array to fail and still maintains integrity of the data. If a program's vulnerabilities are unknowable, then some form of diversity is likely necessary to detect, deter, and otherwise leverage complexity against an attacker. Research is needed to formulate measures of diversity and model it in cyber systems to assess its effectiveness.

Key challenges include

- Developing quantifiable models of fault-tolerant and vulnerability-tolerant architectures
  - Quantifying the benefits of implementation diversity
  - Understanding the scaling of attacker and defender effort in fault-tolerant systems

[2] By fault we mean any anomalous output of the system. All vulnerabilities are faults, but not all faults are vulnerabilities.

- Developing theories and metrics for diversity among variant implementations of the same software, particularly with respect to vulnerabilities (there is the potential to apply this to hardware and networks in modified form)
- Using agent-based models to engineer or discover emergent robustness as a topological property of the graph or network of large collections of cyber subsystems
- Using the above to suggest programming models and network/computer architectures that are inherently more secure

3 Modeling the behavior of programs, machines, and networks

Complex systems pose challenges for a reductionist approach to modeling, with a system's global behavior frequently not being readily deducible from even a detailed understanding of its constituent parts. An additional confounding feature of complex systems is their adaptive nature: Such systems can evolve, often in a coordinated manner, to accomplish system-level imperatives. Such behaviors are said to be "emergent" (they emerge as a consequence of the interactions of the constituents) and being able to reproduce them using appropriate constituent models has often been a matter of trial and error.

3.1 Complex systems modeling

The connections among computers that constitute the Internet, the interactions among virtual components in software systems, and the wiring of logic circuits in microchips can all be represented at an abstract level by graphs and networks. Large numbers of entities with discrete interaction patterns, resulting in unpredictable emergent behavior, form what are known as entity-based complex systems approachable from agent-based or discrete-event algorithms [9, 13]. The emergent behaviors of such systems can, however, be understood and predicted in certain global aspects. Specifically, if we are aware of the impact of these networks and graphs on the emergent behavior of the system, they can be crafted to possibly produce desired emergent behavior like robustness to attack. Because cybersecurity issues are rooted in complexity, modeling that complexity with fidelity is paramount.

Key challenges include

- Understanding the global network dynamics of prototypical complex systems
  - Investigating the emergent behavior of idealized complexity models, such as cellular automata [8, 27] and Boolean networks [15]
  - Extending the results to classify the emergent behavior of more general agent-based models

- Relating realistic software architectures to the results of complexity science, in particular robustness to attack, resilience after attack, etc.
- Designing more general and efficient fault-tolerant architectures using distributed redundancy with emergent robustness, rather than more resource-intensive replication with a single voter
- Predicting the dynamics and spread of malicious software on physical computer networks
- Developing agent-based monitoring and cyber forensic capabilities to detect intrusions based on anomalous dynamics

3.2 Large-scale modeling and simulation

The application of complexity science to real-world cyber systems requires the ability to model and simulate these systems in a rigorous and validated manner. Just as in traditional science and engineering, coarse-grained models that describe a tractable subset of degrees of freedom are crucial for predictive understanding. But due to the undecidability of complex systems, such reduced models can in general describe only overall features, rather than details, of emergent behavior. The validation of such models thus requires particular care. An important avenue for controlled cyber experiments, enabling exploration of emergent behavior and validation of models, is provided by large-scale *emulation*—an especially realistic form of simulation. Emulation allows a physical computer cluster, through creation of numerous virtual machines and virtual network connections, to trace efficiently and with high realism the behavior of a much larger network of computers.

In addition, we need virtual machine managers (VMMs) that allow us to monitor ad-hoc the application's detailed resource access inside the virtual machines in real time, such as network connections, message communication, and CPU/disk/network usage [6]. The advantage of this approach is that the monitoring can be applied transparently to the applications. This monitored data will be part of the emulation or simulation's result set and possibly fed back into our emulation models. Through the same VMMs, we can also enforce fine-grained policies on resource access by the applications, such that we can change and fine-tune our model parameter values in real time, again transparently to the applications themselves.

Key challenges include

- Developing techniques to derive coarse-grained models of complex systems
  - Using renormalization methods from theoretical physics to construct predictive models that preserve key emergent behaviors
  - Applying coarse-grained models to characterize robustness to attack
- Understanding the relations among different levels of modeling and their ability to predict emergent behavior

  - Quantifying tradeoffs between cost and accuracy of models to achieve effective simulations of complex systems
  - Using emergent behavior classifications to guide the choice of appropriate agent-based models

- Extending network emulation/simulation capabilities to Internet scale
- Combining insights from modeling and simulation to offer potential improvements to existing software and network protocols
- Enhancing virtual machine manager implementations to accomodate the detailed monitoring and fine-grained policy enforcements needed for real-time simulation of our models

4 Role of the Department of Energy

The Department of Energy has particular needs not currently being answered by the commercial or research cybersecurity community, as well as unique capabilities to advance the state of the art in cybersecurity research.

4.1 Need for securing DOE's unique open resources

High performance computing (HPC) platforms and attendant software are geared for performance, and any added controls, such as cybersecurity measures, that inhibit performance are unlikely to be adopted. In the Office of Science these platforms and their computational results need to be given the widest possible access to authorized researchers that can be safely achieved. Yet these platforms and attendant simulations are as complex as any in the commercial world and likely heir to the same frailties. Couple this with the fact that some percentage of the university researchers will be using these HPC resources from (unknowingly) compromised platforms and this makes for a particularly challenging problem. Because neither the outside (university researcher's platform) nor the inside (DOE HPC simulation) can be trusted, some sort of virtual machine with sandboxed proxy for the simulation must be considered. The proxy may take the form of a remote method invocation system [3] or even a web-based protocol. The virtual machine within which it executes must enforce a security policy that disallows unsafe operations originating from either side [23]. This policy and attendant enforcement code will benefit from software analysis and formal methods to ensure correctness so that the virtual machine sandbox itself cannot be compromised. More advanced research that allows more flexible and adaptable allocation of potentially dangerous capabilities should be considered as well. The recent, promising research results on creating more secure and safer execution environments for JavaScript and Java, using capability-based techniques and enforced best-practices [17, 19, 20], may be applicable to a virtual machine sandbox environment with fine-grained access control policy enforcement on resource usage and message passing. A wide range of ideas should be considered that enables the DOE-unique environment where both openness is permitted and performance is uninhibited.

4.2 Use of DOE's unique capabilities for cybersecurity research

As discussed previously, large-scale emulation and simulation hold great promise for understanding cybersecurity (Section 3.2). While this is not a new concept [1], DOE is in a unique position to take advantage of it. Within high-performance clusters and Leadership Class Computing (LCC), a virtual Internet can be constructed for emulating and simulating routers, web farms, terminal nodes, etc., and the malefactors that prey on them. Both in scale and speed, LCC machines present an opportunity not found elsewhere for the emulation and simulation of the Internet at a nation-state scale, enabling computer experiments for phenomena that require a global view to understand, such as botnets [7].

Key challenges include

- Researching novel architectures for preserving an open computing and collaboration framework while maintaining a secure environment for DOE-held assets
- Investigating simulation and emulation environments that are enabled by current DOE HPC resources

References

[1] DARPA BAA for Cyber Test Range. http://www.darpa.mil/STO/ia/pdfs/NCR_Qs_and_As.pdf.

[2] RAID. http://en.wikipedia.org/wiki/RAID.

[3] Remote method invocation. http://java.sun.com/javase/technologies/core/basic/rmi.

[4] Rice's theorem. http://en.wikipedia.org/wiki/Rice's_theorem.

[5] Sandbox for computer security. http://en.wikipedia.org/wiki/Sandbox_(computer_security).

[6] sHype: Hypervisor security architecture. http://www.research.ibm.com/secure_systems_department/projects/hypervisor/.

[7] Storm botnet. http://en.wikipedia.org/wiki/Storm_botnet.

[8] P. Bak, C. Tang, and K. Wiesenfeld. Self-organized criticality: An explanation of $1/f$ noise. *Physical Review Letters*, 59:381–384, 1987.

[9] P. Brantley, B. L. Fox, and L. E. Schrage. *A Guide to Simulation*. Springer-Verlag, New York, 1986.

[10] S. S. Brilliant, J. C. Knight, and N. G. Leveson. Analysis of faults in an *N*-version software experiment. *IEEE Transactions on Software Engineering*, 16:238–247, 1990.

[11] J. M. Carlson and J. Doyle. Highly optimized tolerance: A mechanism for power laws in designed systems. *Physical Review E*, 60:1412–1427, 1999.

[12] E. M. Clark, O. Grumberg, and D. A. Peled. *Model Checking*. MIT Press, 1999.

[13] W. J. Dally and B. Towles. *Principles and Practices of Interconnection Networks*, pages 473–509. Elsevier, San Francisco, 2004.

[14] High Confidence Software and Systems Coordinating Group. *High-Confidence Medical Devices: Cyber-Physical Systems for 21st Century Health Care*, pages 23–24. Networking and Information Technology Research and Development Program (NITRD), 2009. http://nitrd.gov/About/MedDevice-FINAL1-web.pdf.

[15] S. A. Kauffman. *The Origins of Order: Self-Organization and Selection in Evolution*. Oxford University Press, 1993.

[16] T. W. Körner. *The Pleasures of Counting*, pages 298–318. Cambridge University Press, Cambridge, UK, 1996.

[17] B. Laurie. Access control. http://www.links.org/files/capabilities.pdf.

[18] B. Littlewood, P. Popo, and L. Strigini. Modeling software design diversity. *ACM Computing Surveys*, 33:177–208, June 2001.

[19] A. Mettler and D. Wagner. The Joe-E language specification (draft). Technical Report UCB/EECS-2006-26, U.C. Berkeley, May 2006. `http://www.truststc.org/pubs/246.html`.

[20] M. S. Miller, M. Samuel, B. Laurie, I. Awad, and M. Stay. Caja: Safe active content in sanitized JavaScript. Technical report, June 2008. `http://google-caja.googlecode.com/files/caja-spec-2008-06-07.pdf`.

[21] J. Oberheide, E. Cooke, and F. Janhanian. CloudAV: *N*-version antivirus in the network cloud. In *Proceedings of the 17th USENIX Security Symposium*, San Jose, CA, July 2008.

[22] D. Quinlan. ROSE: Compiler support for object-oriented frameworks. In *Proceedings of Conference on Parallel Compilers (CPC2000)*, Aussois, France, January 2000.

[23] R. Sahita and D. Kolar. Beyond Ring-3: Fine grained application sandboxing. In *W3C Workshop on Security for Access to Device APIs from the Web*, London, December 2008.

[24] B. Salamat, T. Jackson, A. Gal, and M. Franz. Intrusion detection using parallel execution and monitoring of program variants in user-space. In *Proceedings of EuroSys'09*, Nürnberg, Germany, April 2009.

[25] S. Santhanam, P. Elango, A. Arpaci-Dusseau, and M. Livny. Deploying virtual machines as sandboxes for the grid. In *Proceedings of Second Workshop on Real, Large Distributed Systems*, 2005.

[26] J. R. Sklaroff. Redundancy management technique for Space Shuttle computers. *IBM Journal of Research and Development*, 20:20–28, 1976.

[27] S. Wolfram. *A New Kind of Science*. Wolfram Media, Champaign, IL, 2002.


Sandia National Laboratories