structure in disordered and partially crystalline substances. Beyond fluctuation microscopy, there are related approaches that also look promising (*14*). Revolutions in focusing and brightness make related techniques accessible with penetrating x-rays. Fluctuation microscopy is a fingerprint technique. It is sensitive enough to allow one to distinguish models, but it is difficult to directly interpret data. Further advances will occur by combining fluctuation microscopy data and other structural data in Monte Carlo structural refinements. Progress is needed in the theory underlying interpretation, with the ultimate goal that high-order correlation functions can be directly determined without modeling (*15*). Such developments will provide a better fundamental understanding of amorphous materials and crystal nucleation, resulting in better phase-change memory and other technologies.

References and Notes

1. D. Kaschiev, *Nucleation: Basic Theory with Applications* (Butterworth-Heinemann, Oxford, 2000).
2. U.S. Department of Energy Basic Sciences Advisory Committee, *New Science for a Secure and Sustainable Energy Future* (2008) (www.sc.doe.gov/bes/reports/files/NSSSEF_rpt.pdf).
3. B.-S. Lee *et al.*, *Science* **326**, 980 (2009).
4. I. Peterson, *Sci. News* **130**, 330 (1986).
5. J. T. Jarrett, P. T. Lansbury, *Cell* **73**, 1055 (1993).
6. The artwork of Gordon Halloran (see, for example, www.icepaintingproject.com).
7. D. E. Jesson, M. Kastner, B. Voightlander, *Phys. Rev. Lett.* **84**, 330 (2000).
8. K. F. Kelton *et al.*, *Phys. Rev. Lett.* **90**, 195504 (2003).
9. M. M. J. Treacy, J. M. Gibson, *Acta Crystallogr.* **A52**, 212 (1996).
10. M. M. J. Treacy *et al.*, *Rep. Prog. Phys.* **68**, 2899 (2005).
11. J. M. Gibson, M. M. J. Treacy, *Phys. Rev. Lett.* **78**, 1074 (1997).
12. S. Ovshinsky, *Phys. Rev. Lett.* **21**, 1450 (1968).
13. S. Raoux, *Annu. Rev. Mater. Res.* **39**, 25 (2009).
14. P. Wolchner *et al.*, *Proc. Natl. Acad. Sci. U.S.A.* **106**, 11611 (2009).
15. T. Iwai *et al.*, *Phys. Rev. B* **60**, 191 (1999).

10.1126/science.1182817

COMPUTER SCIENCE

Reflections on Cybersecurity

The lack of security in cyberspace may be addressed by learning from the strengths of the Internet.

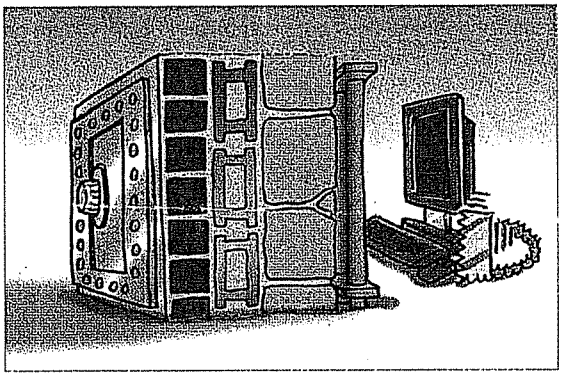
William A. Wulf and Anita K. Jones

> Perfection is achieved, not when there is nothing more to add, but when there is nothing left to take away.
>
> –Antoine de Saint-Exupéry
> in *The Little Prince*

Cyberspace is less secure than it was 40 years ago. That is not to say that no progress has been made—cryptography is much better, for example. But more vital information is accessible on networked computers, and the consequences of intrusion can therefore be much higher. A fresh approach is needed if the situation is to improve materially.

The prevailing assumption continues to be that if systems were implemented correctly, the problem would be solved. Yet, software engineers have tried to do that for 40 years and have failed. A 1993 report from the Naval Research Laboratory (*1*) points to a deeper problem. It analyzed some 50 security breaches, and found that in 22 of those cases, the code correctly implemented the specifications—it was the specifications that were wrong. They handled the usual cases just fine, but did not appreciate that under some circumstances, permitted actions or outcomes were, in fact, security breaches.

A natural tendency is to declare a crisis and convene task forces and an army of programmers to "fix" the security problem(s). But, as detailed in Fred Brooks' *The Mythical Man-Month* (*2*), trying to get more "man months per calendar month" can actually make the situation worse, not better. We conjecture that a similar phenomenon is occurring for cybersecurity. The security model has remained the same since the 1960s, and software engineers have added more and more patches and widgets to try to enforce that security model. The complex interaction of this additional code with the extant code just provides more opportunities for security failures. The cybersecurity community must thus ask whether the problem has been formulated in the right way.



The current model for most cybersecurity is "perimeter defense": The "good stuff" is on the "inside," the attacker is on the "outside," and the job of the security system is to keep the attacker out. The perimeter defense model is built deeply into the very language used to discuss security: Hackers try to "break in," "firewalls" protect the system, "intrusion" must be detected, etc. But is perimeter defense the right underlying model?

We do not think so, for several reasons. First, perimeter defense does not protect against the compromised insider. The Federal Bureau of Investigation (FBI) has reported that in one sample of financial systems intrusions, attacks by insiders were twice as likely as ones from outsiders—and the cost of an intrusion by an insider was 30 times as great (*3*).

Second, it is fragile; once the perimeter has been breached, the attacker has free access. Some will say that this is why "defense in depth" is needed—but if each layer is just another perimeter defense, all layers will have the same problems.

Third, and most important, it has never worked. It did not work for ancient walled cities or for the French in World War II (at 20 to 25 km deep, the Maginot Line was the most formidable military defense ever built, yet France was overrun in 35 days). And it has not worked for cybersecurity. To our knowledge no one has ever built a secure, nontrivial computer system based on this model.

So, what might be an alternative approach? We think we should take our cue from the Internet. That is, there should not be just one model. Rather, there should be a minimal central mechanism that enables implementation of many security policies in application code—systems attuned to the needs of differing applications and organizations.

It is worth noting that the Internet succeeded so well precisely because it does so little. At its core, the TCP/IP protocols, all the Internet does is to promise "best effort" message delivery. It does not promise that the messages will arrive in the order in which they were sent, that they will ever arrive at all, or even that the same message will not

Department of Computer Science, University of Virginia, Charlottesville, VA 22903, USA. E-mail: wulf@virginia.edu, jkj@virginia.edu

arrive multiple times. All of the "smarts" of the net are at its periphery and embedded in "end-to-end" protocols (*4*) that are defined by applications.

Dave Parnas, one of the early software engineers, made a provocative and, we think, deeply important observation that helps to explain the success of the TCP/IP protocols. He pointed out that, when doing a design, the hardest decision to change is the one you make first, because all the subsequent ones to some extent depend on it (*5*). The decision for the TCP/IP protocols to do so little never had to be reconsidered, because it precluded so little.

Is there an analogy to the Internet message delivery design for security? Is there some minimal mechanism that would allow the construction of arbitrary end-to-end security protocols and allow an arbitrary number of these security protocols to coexist simultaneously? Is there a mechanism so simple that, while adequate to support the construction of security policies, does not preempt any decisions on the definition of security or how it is achieved? We think the answer is yes.

But why build multiple "end-to-end security protocols" rather than one really good one? We offer three reasons. First, different applications have different security needs: The requirements of law enforcement emphasize the integrity of the trail of evidence, the intelligence community is most concerned with disclosure of sources and methods, legitimate access to electronic medical records may change dramatically in emergencies, and so on. The point is that desirable security policy is a natural extension of the application; there is no single security policy that serves all needs equally well.

Second, multiple security protocols ensure that if one is broken, the others are not, or at least not in the same way. The current Internet clients form a predominantly Wintel/Cisco monoculture, so a single flaw can make almost the entire net vulnerable to the same attack. Incorporating multiple security policies and multiple implementations of the same policy can dramatically reduce this monoculture-induced vulnerability.

Third, the requirements of future applications cannot be predicted. In the same way as user-defined, end-to-end communications protocols allowed new applications that were not anticipated (such as the Web, search engines, and e-commerce), application-defined security protocols could accommodate unanticipated security requirements.

The lack of cybersecurity has been a consistent concern for 40 years. From time to time that concern flares up, and society resolves to "try harder," but the number of intrusions and their cost have only increased exponentially. It is time to reexamine the basic assumptions, like perimeter defense. Systems based on those assumptions have consistently failed. At least one alternative is an Internet-like minimal mechanism that enables application-defined security definitions.

Is such a minimal mechanism feasible? We think so. In particular, at the network level, an application can use any computable function to decide whether or not to provide its service to a client if it can be absolutely certain who is requesting it. There is a class of algorithms known as "cryptographic protocols" for doing this that require knowing the public key of an object—so we conjecture that by providing just a way of accessing the public key of an object, one could build an arbitrary end-to-end security policy.

References and Notes

1. "A Taxonomy of Computer Program Security Flaws, with Examples," Naval Research Laboratory Report, NRL/FR/5542–93/9591, November 1993.
2. F. Brooks, *The Mythical Man-Month* (Addison-Wesley, Reading, MA, 1975).
3. Testimony of Keith Lourdeau, Deputy Assistant Director, Cyber Division, FBI, before the Senate Judiciary Subcommittee on Terrorism, Technology, and Homeland Security, February 2004.
4. J. H. Saltzer *et al.*, *ACM Trans. Comput. Syst.* 2, 277 (1984).
5. D. L. Parnas, *Commun. ACM* 15, 1053 (1972).

10.1126/science.1181643

DEVELOPMENTAL BIOLOGY

Strategies to Get Arrested

Two gene families in the worm control survival strategies in response to stressful environmental conditions.

Akira Ogawa and Ralf J. Sommer

From bacteria to vertebrates, organisms can respond to changing environmental conditions by arresting their development. Animals in particular have invented a repertoire of diapause programs. As the environment can change at any step of an organism's life cycle, many independent strategies have evolved even within one species. Studies in the nematode *Caenorhabditis elegans* are beginning to show not only the diversity of these strategies, but also the genetic and genomic mechanisms mediating the response. On pages 994 and 954 of this issue, Kim *et al.* (*1*) and Angelo and Van Gilst (*2*) reveal how members of two multigene families—nuclear hormone receptors and G protein–coupled receptors—perceive and translate environmental cues to regulate diapause stages in the larval and adult reproductive stages, respectively.

C. elegans became a model organism in part because of the ease with which it can be cultured in the laboratory. On petri dishes with *Escherichia coli* as food source, this animal can complete its life cycle in as little as 3 days (see the figure). However, this is only observed when food is unlimited, a scenario that is unrealistic in the natural world. Not surprisingly, therefore, recent studies suggest that in nature, *C. elegans* follows a different path. Animals are most often found in the so-called dauer stage, a developmentally arrested stage (*3*). Lab-based studies revealed that the dauer stage occurs when larvae have little food or are exposed to high temperature or a high concentration of dauer pheromone, which is secreted constitutively by the members of a population (*4*). Although the existence of a pheromone was shown more than three decades ago, only recently have studies characterized it as a complex mixture of chemicals. Ascarosides, a class of glycosides with a dideoxysugar moiety and variable side chains, regulate entry of larvae into the dauer phase and also social behaviors in adults (*5*–*8*). Genetic studies have identified signaling systems involved in dauer regulation, including insulin and transforming growth factor–β signaling (*9*). However, how the dauer pheromone is sensed and how it is coupled to signal transduction have not been clear.

Kim *et al.* report that two chemoreceptors that are G protein–coupled receptors—*srbc-64* and *srbc-66*—mediate the effects of the dauer pheromone. When these two receptors, which are expressed in a pair of sensory neurons, were mutated, responses to ascarosides were impaired. However, a nematode strain carrying a mutation in both genes still retained some responsiveness to ascarosides, indicating that dauer pheromone perception involves multiple receptors. The results also reveal unexpected complexity in both pheromone production and sensing.

Max Planck Institute for Developmental Biology, 72076 Tübingen, Germany. E-mail: akira.ogawa@tuebingen.mpg.de; ralf.sommer@tuebingen.mpg.de