NSF/IARPA/NSA Workshop on the Science of Security

David Evans (PI), *University of Virginia*

Workshop Report

The *NSF/IARPA/NSA Workshop on the Science of Security* was held 17-18 November 2008 in Berkeley, CA. It brought together a group of 43 leading researchers in computer security and a wide variety of other relevant fields to consider the state of scientific research in computer security and to identify steps toward establishing a stronger scientific basis for computer system security (see Section 4 for a list of participants).

This report summarizes the recommendations emerging from the workshop and the views presented there. Slides from most of the full presentations are available on the workshop website, *http://sos.cs.virginia.edu/agenda.html*. The opinions and ideas in this report are meant to reflect the views of participants in the workshop, but have not been reviewed or approved by all the participants.

1 Executive Summary

The attendees reached a strong consensus that while there is some scientific work in computer security, our community would benefit from efforts to provide a stronger scientific basis for computer security. This was considered from three senses of the meaning of *science*:

1. The weak sense, of science as the *systematization and generalization of knowledge*. There have been many results in recent years that have provided strong understanding of a particular vulnerability or the security issues involved in designing a given system, but little effort or success in collecting knowledge resulting from these works into a general and systematic framework. There was strong agreement that computer security would benefit from efforts in this direction. One goal (suggested by Fred Schneider) is to establish a common framework for classes of defenses, categorized according to the policies that they can enforce and the classes of attacks those policies can thwart.

2. The strong sense, of science as a way to develop *universal laws* that can be used to make strong, quantitative predictions. This is more controversial since although everyone agrees that such laws would be wonderful to have, there is no clear consensus whether or not such laws exist for system security. Nevertheless, efforts directed towards finding such laws would be useful, even if they lead to a clearer understanding of why such universal laws do not exist.

3. The methodological sense, of science as a way to conduct research by following the *scientific method* of forming hypotheses and carrying out experiments. For certain areas of computer security, experiments seem useful and the community will benefit from better experimental infrastructure, data sets, and methods. For other areas, it seems difficult to do meaningful experiments without developing a way to model a sophisticated and creative adversary.

From this perspective, three areas were identified as the most promising directions for research leading towards a stronger scientific basis for computer security: metrics, formal methods, and experimentation. A common theme across all three areas is the challenge of reconciling the need to make assumptions to enable reasoning about systems with the challenge posed by creative adversaries who attempt to exploit those assumptions. In other fields, mistaken assumptions may lead to inaccurate results; in computer security, every assumption is a potential security vulnerability.

1.1 Metrics

Despite Lord Kelvin's oft-repeated maxim that one cannot really claim to understand something until it can be measured, few meaningful tools exist for measuring the security of a computing system. Without such tools, it is difficult to measure progress scientifically, and to make practical decisions. Metrics can be either analytical or experimental, and there is a need to develop and explore the value of both types of metrics for computer security.

The fundamental challenge for any metrics in computer security is that metrics, by their nature, require assumptions and abstractions. The goal of a metric is to take a complex system and produce a scalar value that describes some important property of that system. In the case of computer security, any assumptions embedded in a metric are potential security vulnerabilities. In some sense, what we really need is a *meta-metric* — a way to measure the risks associated with all the assumptions used to define the metric. One interesting direction is to consider changing the design approach to focus on designing for measurable security properties. Another idea for making progress towards useful metrics included devising challenge problems to investigate metrics.

Computational Complexity Metrics. The field of cryptology has built a strong theoretical foundation over that past two decades by defining classes of adversaries based on the amount of information (e.g., Chosen Ciphertext Attacker) and computational power (e.g., polynomial-time adversary) available to them. In some cases, computational complexity metrics seem promising. In cryptology, certain ciphers are argued to be strong since breaking them requires solving a problem that is believed to be hard (e.g., RSA and factoring). Extending this approach to system security seems worthwhile, but poses a number of challenges. The first problem is the difference between mathematical abstractions and real implementations. The gap between theoretical cryptography results and practical cryptanalysis illustrates this: although no one has found a fast factoring algorithm, RSA implementations are regularly broken because of side channels (such as timing and power consumption), poor random number generation, insecure key storage, and programming bugs. For system security, the gap between models simple enough to use for metrics and actual implementations is even larger. To make progress, we need metrics that work on less abstract models of actual systems, or ways to build systems that refine models without introducing new security vulnerabilities. The second problem with security metrics is the difficulty in reasoning about creative adversaries. Since it seems unlikely to reason well about adversary creativity, this argues for metrics that assume adversaries can efficiently search the entire space of possible actions, but possibly complexity metrics can be used to analyze that space and the maximum effectiveness of different types of search strategies. For example, some security techniques based on automatically generated diversity can be analyzed according to the computational and communication resources available to the adversary.

Economic and Biological Metrics. An alternative to computational complexity metrics are metrics based on economic or biological approaches. A research community has emerged around economic approaches to security (embodied by the annual Economics of Information Security workshops), and metrics based on the expected cost required to compromise a system can provide useful guidance to system designers, but for most computing systems the parameters for these metrics are somewhat arbitrary. Metrics for network vulnerability can be developed based on epidemiology, but depend on abstract models of network nodes.

Experimental Metrics. In addition to analytical metrics, there is also value in developing experimental metrics. An analogy is to the way mechanical safes are rated based on their ability to withstand different classes of attacks for a given time period (e.g., a TL-30 safe is rated to be able to withstand 30 minutes of attacks involving various mechanical and electrical tools). Red team exercises are sometimes used to evaluate the security of a system, but current red team exercises are too ad hoc to produce a meaningful measurement beyond knowing how easily a particular red team was able to compromise a given system. Instead, they are primarily used to identify apparent weaknesses in a particular deployment. Perhaps more systematic approaches to adversarial metrics can be developed, although it seems difficult to overcome the problem of variations in attacker creativity.

1.2 Formal Methods

Formal methods have made great strides in reasoning about correctness properties of software systems. Two trends argue for optimism in usefully applying formal methods to system security: (1) the capability of formal techniques to scale to large systems is increasing, and (2) developments in secure processors and software architectures for secure systems continue to reduce the size of the trusted computing base needed to establish the security of a complex system. The other main issue is determining what properties to prove to establish useful security properties. This requires development of modal logics or other techniques that can better model what designers want to know about systems.

Shrinking the Trusted Base. Systems continue to get more complex, but progress has been made towards architectures that shrink and isolate the security-critical parts of the system. Virtualization is becoming prevalent, and there is much promising work in designing and using virtual machines for security. Separation kernels carry this isolation even further. More work needs to be done to develop VMs that provide provable isolation properties. One challenge is VM-based security is reasoning about events at the level of the guest OS, while using observations at the level of the host OS. The presents a large semantic gap between the low-level events visible to the host OS, and the high-level events for which we want to design security policies, so it is necessary to develop mechanisms to bridge this gap and enable high-level policy enforcement at the VM level.

Composition. Another important area is understanding how to reason about security properties of when components are composed. We have many tools available for reasoning about isolated components, but typically need to resort to worst-case assumptions when reasoning about interactions. We need to develop ways of reasoning about composed systems that preserve abstraction boundaries without needing to rely on worst-case assumptions, and ways to develop and compose components such that the security properties of compositions of those components can be derived from established properties of the components. We also need richer mechanisms than just complete isolation for reasoning about interactions between components.

Reducing the Gap between Models and Implementations. Most formal methods rely on abstract models of the target system. In security arguments, any assumptions used to construct the abstract model are potential security vulnerabilities. Promising directions include refinement approaches where a model becomes increasingly detailed and realistic, and formal methods that scale well enough to work directly on source code or executables. Another important direction is finding ways to make all assumptions used to make a formal argument about a real implementation explicit.

1.3 Experimentation

The main challenge in doing meaningful experiments in computer security is the need to model an adversary. Two main directions for work are suggested: finding ways to improve our adversary models, and finding ways to design more reproducible experiments that are not so dependent on accurate models of adversary behavior.

Improving Adversary Models. One suggested approach to improving adversary models for experiments is to systematize current knowledge about real adversaries and use that as a basis for developing experimental adversary models. Perhaps such an approach can be used to develop a canonical attacker model that could be used in a wide class of experiments, rather than relying on *ad hoc* models created by individual experimenters. In general, however, it seems extraordinarily difficult to evaluate new security mechanisms using experiments; there is little cause for optimism that a useful model of a creative adversary attacking a new design can be developed.

Design for Reproducibility. One weakness in empirical security research is that few experiments are ever reproduced. Some of this results from the lack of incentives to academic researchers to perform this type of work. One suggestion is to have students reproduce published experimental work in early graduate courses. A more fundamental problem is that many security experiments are not designed in a way that makes them readily reproducible. Better sharing of code and data sets is also important.

2 Agenda

Monday, 17 November 2008

9:00am Welcome and Introduction
Karl Levitt, National Science Foundation
Lisa Porter, Director, Intelligence Advanced Research Projects Activity
Frederick Chang, University of Texas at Austin
and former Director of Research, National Security Agency

9:30-10:30 Fred B. Schneider, Department of Computer Science, Cornell University
A Map For Security Science

11-noon Panel: *Is there a science of security (and if so, what might it look like)?*
Moderator: Carl Landwehr, IARPA
Panelists:
Anupam Datta, Carnegie Mellon University
Joshua Guttman, MITRE
Michael Reiter, University of North Carolina

1:30-2:30 Panel: *What can we learn from other fields?*
Moderator: Cliff Wang, ARO
Panelists:
Stephanie Forrest, University of New Mexico
Alfred Hero, University of Michigan
Stuart Russell, University of California, Berkeley

3:00-4:15 Breakout group discussions:
What can we learn from other fields?
Leader: Pierre Moulin, UIUC
How can we design systems with known security properties?
Leader: Rebecca Wright, Rutgers University
Is there a scientific way to measure security?
Leader: Shouhuai Xu, University of Texas at San Antonio

Tuesday, 18 November 2008

9-9:30am Frederick Chang, University of Texas at Austin and
former Director of Research, National Security Agency

9:30-10:30 Panel: *How can we reason about impossible things?*
Moderator: Robert Herklotz, AFOSR
Panelists:
Byron Cook, Microsoft Research Cambridge/Cambridge University
Dusko Pavlovic, Kestrel Institute/Oxford University
Hal Varian, University of California, Berkeley/Google

10:45-11:15	John Doyle, Professor of Control & Dynamical Systems, Electrical Engineering, and BioEngineering, California Institute of Technology
11:30-12:30	Panel: *Are scientific experiments in security possible?*
	Moderator: Karl Levitt, NSF
	Panelists:
	Roy Maxion, Carnegie Mellon University
	John Mitchell, Stanford University
	Vicraj Thomas, BBN Technologies
1:30-2:30	Breakout discussions:
	Complexity
	Leader: Sampath Kannan, NSF
	Experimentation
	Leader: Karl Levitt, NSF
	Composition
	Leader: John Rushby, SRI
2:30-3:00	Summary discussion
3:15-4:15	Breakout discussions: *Questions and Promising Approaches for a Science of Security*
4:15-5:00	Discussion, wrap-up

3 Working Questions

The workshop started with a set of working questions that a science of security should be able to answer. Refined versions of those questions are below.

Attack Models. Modern cryptography has developed a set of formal attack models that precisely describe attacker capabilities and enable formal reasoning about the strength of cryptographic algorithms. Are there analogous formal attack models for computer systems? Can we use them to reason formally about the resilience of systems to attackers with different capabilities?

System Resilience. Given a system P and an attack class A, is there a way to:

1. Prove that P is not vulnerable to any attack in A?
2. Construct a system P' that behaves similarly to P except is not vulnerable to any attack in A? (This requires a clear notion of what *behaves similarly* means.)

Metrics. Given two systems, are there meaningful ways to compare their security. For example,

1. Can we define formally what more secure means?
2. Are there meaningful quantitative measures of the security of a system?
3. How can we determine scientifically if system A is more or less secure than system B?

4. How can we determine scientifically if system A is more secure than system $f(A)$ (where f is some function that transforms an input program or system definition)?

Composition. Given a set of components with known properties, is there a way to assure and reason about security properties of a system composed of these components? Is it possible to design components that behave securely regardless of how they are composed?

Refinement. Formal methods have developed refinement techniques where correctness properties can be reasoned about as abstract designs are transformed through successive steps into concrete implementations. Is there a way to go from a design to an implementation with assurance that the implementation preserves security properties of the design? Is there a way to prove that a particular change to a system makes it more secure in some sense, without making it less secure in any other sense?

Program Analysis. What are the fundamental limits of each dynamic and static analysis with respect to establishing security properties? How can these limitations be overcome using hybrid analysis or non-universal programming languages?

Worst-Case Assumptions. Current approaches attempt to divide systems into completely untrusted components and a minimal trusted computing base (TCB) on which all security properties depend. It seems too difficult in most systems to make the TCB small enough to be completely secure, but also overly weak to assume nothing about other components. Are there ways to design systems with less absolute requirements for components, and a many-gradation division between trust levels?

4 Attendees

Organizers

David Evans, *University of Virginia*
Karl Levitt, *National Science Foundation, Program Manager*
Brad Martin, *National Security Agency*
James Silk, *Institute for Defense Analyses*

Participants

David Bray, *Institute for Defense Analyses*
Frederick Chang, *University of Texas at Austin*
Byron Cook, *Microsoft Research Cambridge/Cambridge University*
Son Dao, *HRL Laboratories*
Anupam Datta, *Carnegie Mellon University*
John Doyle, *California Institute of Technology*

Anthony Ephremides, *University of Maryland*
Manfai Fong, *Intelligence Advanced Research Projects Activity*
Stephanie Forrest, *University of New Mexico*
John Frink, *Office of the Deputy Under Secretary of Defense*
Joshua Guttman, *The MITRE Corporation*
Robert Herklotz, *Air Force Office of Scientific Research, Program Manager*
Alfred Hero, *University of Michigan*
Sampath Kannan, *National Science Foundation, University of Pennsylvania*
Steven King, *Office of the Deputy Under Secretary of Defense*
Carl Landwehr, *Intelligence Advanced Research Projects Activity*
Greg Larson, *Institute for Defense Analyses*
John Mallery, *Massachusetts Institute of Technology*
Robert Meushaw, *National Security Agency*
Roy Maxion, *Carnegie Mellon University*
John Mitchell, *Stanford University*
Pierre Moulin, *University of Illinois at Urbana-Champaign*
Tim Murphy, *Intelligence Advanced Research Projects Activity, Deputy Director*
Dusko Pavlovic, *Kestrel Institute and Oxford University*
Nick L. Petroni, Jr., *Institute for Defense Analyses Center for Computing Sciences*
Lisa Porter, *Intelligence Advanced Research Projects Activity, Director*
Michael Reiter, *University of North Carolina*
Phillip Rogaway, *UC Davis*
John Rushby, *SRI International*
Stuart Russell, *UC Berkeley*
Fred B. Schneider, *Cornell University*
Dawn Song, *University of California, Berkeley*
Pieter Swart, *Los Alamos National Laboratory*
Vicraj Thomas, *BBN Technologies, GENI Project Office*
Hal Varian, *UC Berkeley and Google*
Cliff Wang, *Army Research Office*
Rebecca Wright, *Rutgers University*
Shouhai Xu, *University of Texas at San Antonio*
Ty Znati, *National Science Foundation*
Lenore Zuck, *National Science Foundation*