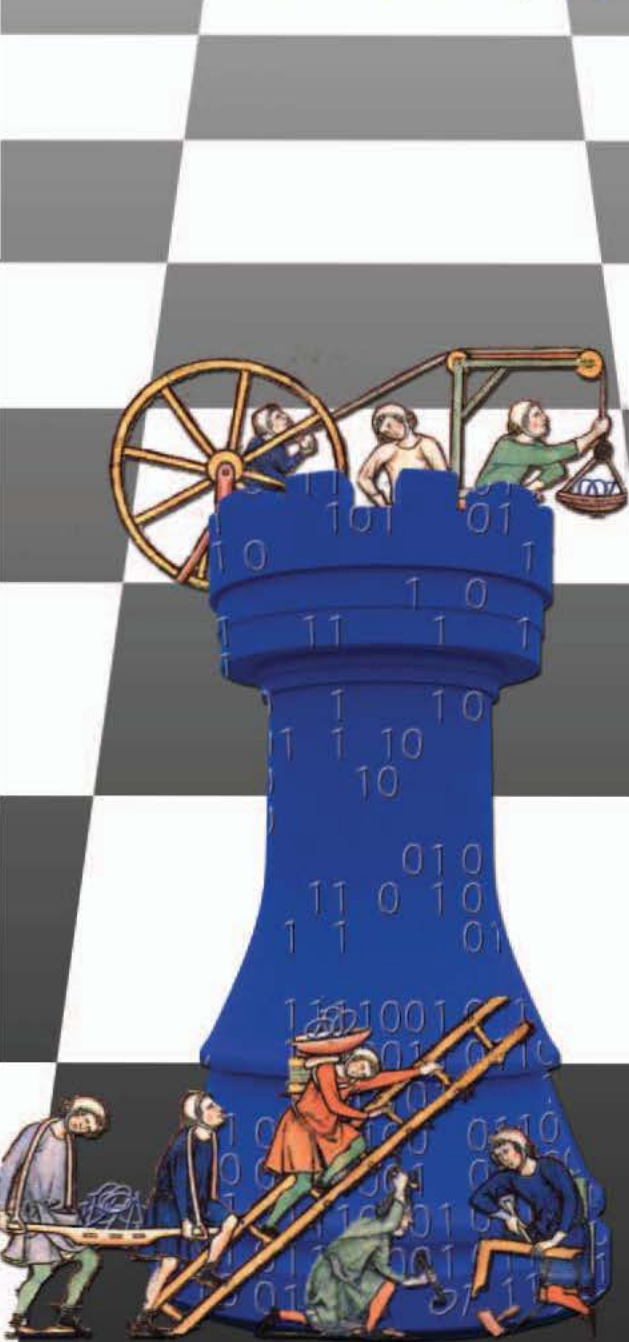




# THE Next Wave

The National Security Agency's review of emerging technologies



Building a science of cybersecurity:  
The next move



Globe at a Glance | Pointers | From Lab to Market





# THE Next Wave

The National Security Agency's review of emerging technologies

## GUEST Editor's column

Stuart Krohn

In 2012, *The Next Wave (TNW)* devoted two issues to cybersecurity, focusing on the need for basic scientific foundations to underpin our trust in systems. Creating this “Science of Security” (SoS) requires sustained support and long-term vision. One only needs to look at the news headlines to observe that the need for more secure systems continues to exist and our efforts are far from over.

The good news, I believe, is that we are beginning to see the emergence of some components necessary for a true science—including a community of cybersecurity researchers whose methods are rigorous, who are willing to challenge, refute, and improve ideas, and who are providing results that can be built upon by others. In addition, more and more universities are adding undergraduate and graduate courses or units that examine scientific rigor in security research.

Of course, it is essential that this new science be grounded on common definitions. Throughout the years, there has been much debate about the nature of science—what it is, and what methods are best. The works of Karl Popper on falsifiability, of Pierre Duhem on the testing of hypotheses as parts of whole

bodies of theory, and of Thomas Kuhn on shifts in scientific paradigms, are fine examples of this. No doubt we will continue that broader discussion in relation to security science, but that is not our interest here.

---

**It is essential that this new science be grounded on common definitions.**

---

This issue of *TNW* describes research contributing to the development of security science. Included are highlights of two workshops initiated by the Special Cyber Operations Research and Engineering subcommittee: one on the adoption of cybersecurity technology, the other on computational cybersecurity in compromised environments. We also present highlights of the ongoing multidisciplinary university research at the SoS lablets. Interspersed are several more in-depth papers on topics including power grid security, phishing, privacy, cyber-physical systems, and a competition aimed at building better code.

# Contents



Whether you are a long-time reader of *TNW* or have just discovered it, we hope you will always count on this publication as a “go to” source for news on emerging technologies.



Stuart Krohn  
Technical Director, Science of Security  
Strategic Engagements, NSA

[Cover photo credits: pialhovik, princessmaro/iStock/Thinkstock, Wavebreakmedia Ltd/Wavebreak Media/Thinkstock, Marie Reed]

- 2 Resilient and secure cyber-physical systems  
WILLIAM EMFINGER, PRANAV SRINAVAS KUMAR,  
GABOR KARSAI
- 8 Improving power grid cybersecurity
- 12 Analyzing the cost of securing control systems  
ZHENQI HUANG, YU WANG, SAYAN MITRA,  
GEIR DULLERUD
- 18 Build it, break it, fix it: Competing to build secure systems  
MICHAEL HICKS, ANDREW RUEF
- 24 The social engineering behind phishing  
CHRISTOPHER B. MAYHORN, EMERSON  
MURPHY-HILL, OLGA A. ZIELINSKA,  
ALLAIRE K. WELK
- 32 GLOBE AT A GLANCE: NSA Science of Security research network
- 34 POINTERS: Science of Security research and events highlights
- 41 FROM LAB TO MARKET: NSA shares cyber software via open source

*The Next Wave* is published to disseminate technical advancements and research activities in telecommunications and information technologies. Mentions of company names or commercial products do not imply endorsement by the US Government.

This publication is available online at <http://www.nsa.gov/research/tnw/tnw211/article1.shtml>. For more information, please contact us at [TNW@tycho.ncsc.mil](mailto:TNW@tycho.ncsc.mil).



# Resilient and secure cyber-physical systems

William Emfinger | Pranav Srinivas Kumar | Gabor Karsai







Computers are increasingly ubiquitous in the world as researchers are developing networks of smart devices that work together. Smart devices—such as autonomous cars, smart parking lots, smart houses, and smart traffic systems—are the building blocks to the smart grid, which monitors and controls these devices and their information. This enables them to provide services such as faster and safer commuting, greater energy efficiency, and remote monitoring for users. Such networks of smart devices, which sense and react to the world, are classified as cyber-physical systems.

## Cyber-physical systems

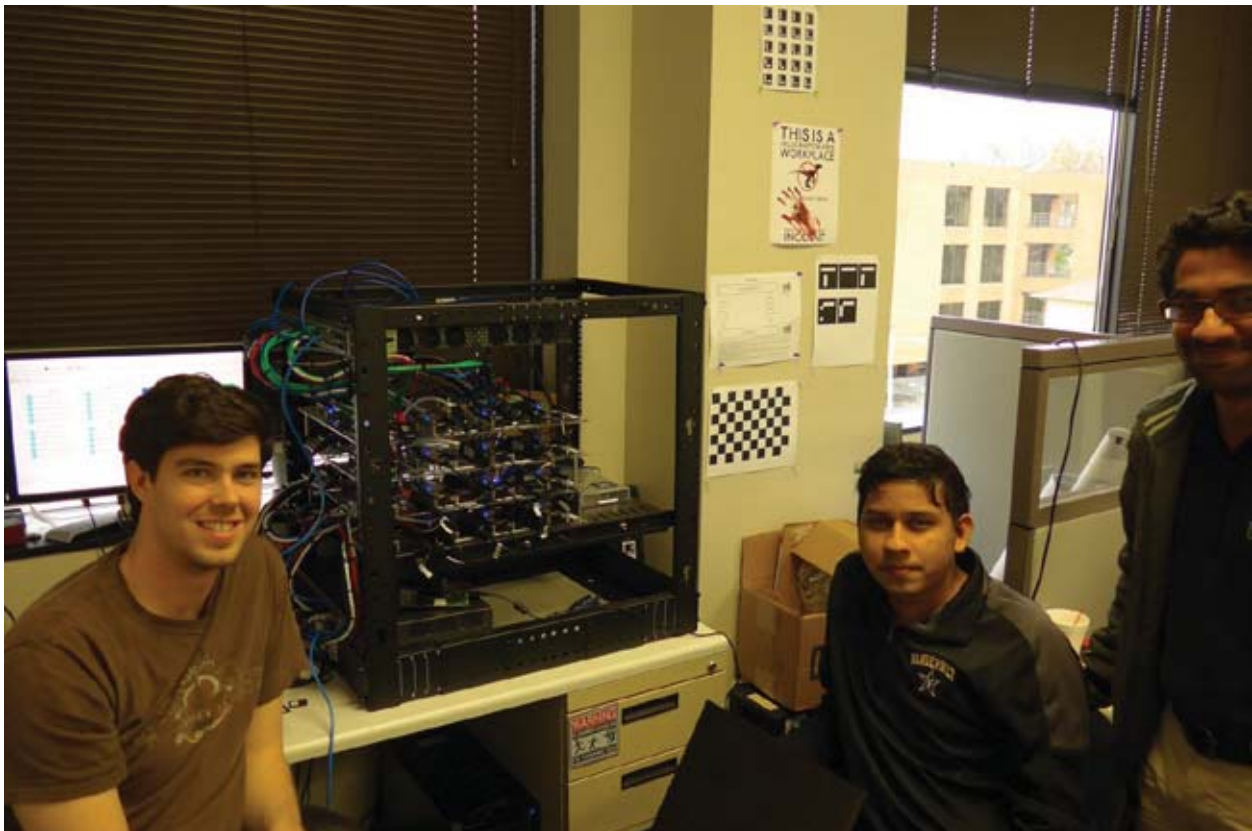
A cyber-physical system (CPS) is composed of embedded computing nodes communicating and collaborating to control aspects of its physical environment. The interaction between the computers and their environment, however, causes a range of complexities—in development, testing, verification, run-time, and management—which must be properly handled. These complexities compound the already difficult task of ensuring reliability and security of the CPS as a whole.

An example of such a CPS is a fractionated (i.e., divided) satellite cluster orbiting in formation. Each satellite provides communications, sensing, and computational resources to the rest of the cluster in fulfillment of the mission goals. It is difficult to develop the satellite system and the applications which control the system and run on it; this necessitates the use of system and software models for design-time analysis and verification of system performance and stability.

Such analysis includes the verification of operation deadline and timing properties, as well as network resource (e.g., buffer capacity and bandwidth) characteristics provided by the system and required by the applications running on the system. However, for such safety-, mission-, and security-critical systems, we must further ensure that the system is resilient to faults and anomalies. It must also be secure against attacks from compromised components within the system as well as from external sources.

## CPS development process

To facilitate rapid and robust design and development of applications for our resilient CPS (RCPS) test bed (see figure 1), we developed an integrated tool suite for model-driven system and application development. Using this tool suite, we can describe—very precisely—the component-based software architecture for the applications which will run on the CPS in



**FIGURE 1.** This resilient cyber-physical systems (RCPS) test bed, developed by William Emfinger (left), Pranav Kumar (center), and Amogh Kulkarni (right), performs security and resilience testing of CPS and their applications.

service of the mission's goals. These software models are developed in tandem with system models that incorporate the physics of the system. (For example, network characteristics such as bandwidth and latency between satellites vary periodically as functions of time according to the satellites' orbital parameters).

By composing these models together, we can analyze at design time the resource requirements and utilization of the applications on the system. Furthermore, the code generators we developed for the tool suite allow us to generate most of the application code (i.e., the infrastructural code which interfaces with the middleware) in a reliable manner. By relying on these code generators, we are able to focus on the core business-logic code, which provides the functionality we want from the applications on the cluster. These applications allow us to test the systems we are examining; for example, testing the detection and mitigation strategies for compromised or malicious software components based on the behavior of their network traffic.

## Resilient CPS test bed

The RCPS test bed itself (see figure 2) is composed of the following components:

- ▶ **32 embedded Linux computers** (BeagleBone Blacks) with ARMv7L architecture;
- ▶ **OpenFlow-capable smart gigabit network switch**, which allows the network characteristics of the system to be enforced on all network traffic;
- ▶ **physics simulation**, which allows the physical dynamics of the hardware to be simulated along with the sensor data and actuator control (for our satellite cluster system, we use Orbiter Space Flight Simulator);
- ▶ **standard gigabit network switch**, which allows fast communication (simulating the hardware bus) between the physics simulation and the nodes of the cluster; and
- ▶ **development machine**, which allows the modeling, development, deployment, and monitoring of the application code which runs the system.

By integrating the physics simulation and network emulation into the cluster (see figure 3), we are able to, in the case of the satellite cluster example, use the



**FIGURE 2.** This RCPS test bed contains 32 embedded Linux computing boards. Each board (middle) is connected to both a smart network switch (top) which performs network emulation using OpenFlow and a regular network switch (bottom) that provides access to the physics simulation (Orbiter Space Flight Simulator).

physics simulation to determine the network characteristics between the nodes of the cluster. We can then enforce those characteristics (i.e., bandwidth, delay, and packet loss) on the cluster's network traffic through the smart switch. In this way, we can ensure that the applications running on the cluster see the same sensor and network behavior as they would in the real system. Because these types of mobile, networked CPSs are becoming more prevalent, the






**FIGURE 3.** This Orbiter Space Flight Simulator simulation of the 32-node satellite cluster, which is controlled by the RCPS test bed, calculates the physics of the satellites and simulates their sensors and actuators.

network resources are becoming more crucial to the systems' functionality. Therefore, the emulation of the network is required to ensure high fidelity of application test results with respect to the run-time system.

We are currently developing test applications which use our research into network and timing analysis techniques to detect malicious software components at run-time and mitigate the effect of their attacks. Using these techniques, our infrastructure will provide a stable system, capable of detecting coordinated attacks from distributed software components (e.g., a denial-of-service (DDoS) attack from compromised or malicious software attempting to bring down a system node or an attack on specific sensors and actuators to make the system unstable).

## Summary

We created the RCPS test bed as the foundational infrastructure for running experiments on CPSs and their software. A critical part of a CPS is the interaction with and feedback from the physical world, so

the integration of the physics simulation increases the fidelity of our cluster test results with respect to the system we are analyzing. The modeling, analysis, generation, deployment, and management tool suite we have developed drastically cuts down on the application development difficulty and time. This allows us to focus on the tests we want to run and the systems we want to analyze. 

## About the authors

**William Emfinger** has a PhD in electrical engineering from Vanderbilt University in Nashville, Tennessee. He received his BS in electrical engineering and biomedical engineering at Vanderbilt in 2011. During his undergraduate studies, Emfinger worked in rehabilitation engineering. His research interests combine cyber-physical/embedded systems engineering with high-performance and high-fidelity system simulation and rendering, with a focus on aerospace systems engineering. Besides these research interests, he enjoys theoretical physics and mathematics. He expects to complete his PhD studies in 2015.

**Pranav Srinivas Kumar** is a graduate research assistant at the Institute for Software Integrated Systems at Vanderbilt University. His research focuses on modeling, analysis, simulation and verification of component-based software applications executed on distributed real-time embedded systems. He received his BE in electronics and communication engineering from Anna University in Chennai, India in 2011.

**Gabor Karsai** is a professor of electrical and computer engineering at Vanderbilt University and senior research scientist at the Institute for Software Integrated Systems. He has over 20 years of experience in software engineering. Karsai conducts research in: the design and implementation of advanced software systems for real-time, intelligent control systems; programming tools for building visual programming environments; and the theory and practice of model-integrated computing. He received his BSc and MSc in electrical and computer engineering from the Technical University of Budapest in 1982 and 1984,

and he received his PhD in electrical and computer engineering from Vanderbilt University in 1988. Karsai has published over 160 papers, and he is the coauthor of four patents.

## Acknowledgements

We would like to thank Dexter Watkins and Cameron Ridgewell for their assistance in the design and development of the RCPS test bed.

*This work was supported by the Air Force Research Laboratory (AFRL) Science of Secure and Resilient Cyber-Physical Systems project under contract FA8750-14-2-0180 and by Vanderbilt University. We thank the sponsors for their generous support. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the AFRL or the US government.*

# Improving power grid cybersecurity



The Information Trust Institute (ITI) ([iti.illinois.edu](http://iti.illinois.edu)) at the University of Illinois Urbana-Champaign (UIUC) has a broad research portfolio in the field of information security and has been researching issues related to electric power infrastructure and the development of a stronger, more resilient grid. Their research efforts have significantly contributed to the Trustworthy Cyber Infrastructure for the Power Grid (TCIPG) project. Ten years ago, the electricity sector was largely “security unaware.” Since then, thanks in part to TCIPG, the industry has broadly adopted security best practices. That transition came about through breakthrough research, national expert panels, and the writing of key documents. Because the threat landscape continuously evolves, however, it is essential to maintain resiliency in a dynamic environment and ensure continuous improvement.



The TCIPG project (<http://tcipgpro.cpanel.engr.illinois.edu>), a partnership among Illinois and three other leading US universities—Dartmouth College, Arizona State University, and Washington State University—as well as governmental and industrial organizations, looks for ways to protect the grid’s underlying computing and communication network infrastructure from malicious attacks as well as from accidental causes, such as natural disasters, misconfiguration, or operator errors. TCIPG participants continually collaborate with the national laboratories and the utility sector to improve the way that power grid infrastructure is designed.

TCIPG comprises several dozen researchers, students, and staff who bring interdisciplinary expertise essential to the operation and public adoption of current and future grid systems. That expertise extends to power engineering; computer science and engineering; advanced communications and networking; smart-grid markets and economics; and science, technology, engineering and mathematics (STEM) education.

### TCIPG research in smart grid resiliency

Countering threats to the nation’s cyber systems in critical infrastructure such as the power grid has become a major strategic objective and was identified as such in Homeland Security Presidential Directive 7 [1]. Smart-grid technologies promise advances in efficiency, reliability, integration of renewable energy sources, customer involvement, and new markets. But to achieve those benefits, the grid must rely on a cyber-measurement and control infrastructure that includes components ranging from smart appliances at customer premises to automated generation control. Control systems and administrative systems no longer have an air gap; security between the two has become more complicated and complex.

TCIPG research has produced important results and innovative technologies in addressing that need and the complexity by focusing on the following areas:

- ▶ Detecting and responding to cyberattacks and adverse events, as well as incident management of these events;
- ▶ Securing of the wide-area measurement system on which the smart grid relies;

- ▶ Maintaining power quality and integrating renewables at multiple scales in a dynamic environment; and
- ▶ Developing advanced test beds for experiments and simulation using actual power system hardware “in the loop.”

Much of this work has been achieved because of the success of the experimental test bed.

### Test bed cross-cutting research

Experimental validation is critical for emerging research and technologies. The TCIPG test bed enables researchers to conduct, validate, and evolve cyber-physical research from fundamentals to prototype, and finally, transition to practice. It provides a combination of emulation, simulation, and real hardware to realize a large-scale, virtual environment that is measurable, repeatable, flexible, and adaptable to emerging technology while maintaining integration with legacy equipment. Its capabilities span the entire power grid—transmission, distribution and metering, distributed generation, and home automation and control. Together, these provide true end-to-end capabilities for the smart grid.

The cyber-physical test bed facility uses a mixture of commercial power system equipment and software, hardware and software simulation, and emulation to create a realistic representation of the smart grid. This representation can be used to experiment with next-generation technologies that span communications from generation to consumption and everything in between. In addition to offering a realistic environment, the test bed facility has cutting-edge research and commercial instruments that can explore problems from multiple dimensions, tackling in-depth security analysis and testing, visualization and data mining, and federated resources, and developing novel techniques that integrate these systems in a composable way. “Composable” means each part of the system is secure and continues to be secure when joined to all the other parts.

A parallel project funded by the state of Illinois, the Illinois Center for a Smarter Electric Grid (ICSEG), is a five-year project to develop and operate a facility to provide services for the validation of information

technology and control aspects of smart-grid systems, including microgrids and distributed energy resources. This project's key objective is to test and validate in a laboratory setting how new and more cost-effective smart-grid technologies, tools, techniques, and system configurations can be used in trustworthy configurations to significantly improve those in common practice today. The laboratory is also a resource for smart-grid equipment suppliers and integrators and electric utilities to allow validation of system designs before deployment.

## Education and outreach

In addition to basic research, TCIPG has addressed needs in education and outreach. Nationally, there is a shortage of professionals who can fill positions in the power sector. Skills required for smart-grid engineers have changed dramatically. Graduates of the collaborating TCIPG universities are well-prepared to join the cyber-aware grid workforce as architects of the future grid, as practicing professionals, and as educators.

## Continuing education

In the area of continuing education, TCIPG:

- ▶ Conducts short courses for practicing engineers and for Department of Energy (DOE) program managers;
- ▶ Holds a biennial TCIPG Summer School for university students and researchers, utility and industry representatives, and government and regulatory personnel;
- ▶ Organizes a monthly webinar series featuring thought leaders in cybersecurity and resiliency in the electricity sector; and
- ▶ Conducts extensive STEM outreach to K–12 students and teachers. (TCIPG has developed interactive, open-ended apps (iOS, Android, MinecraftEdu) for middle-school students, along with activity materials and teacher guides to facilitate integration of research, education, and knowledge transfer by linking researchers, educators, and students.)

The electricity industry in the United States is made up of thousands of utilities, equipment and software vendors, consultants, and regulatory bodies. In both its National Science Foundation (NSF)-funded and DOE/Department of Homeland Security (DHS)-funded phases, TCIPG has actively developed extensive relationships with such entities and with other researchers in the sector, including conducting joint research with several national laboratories.

The involvement of industry and other partners in TCIPG is vital to its success and is facilitated by an extensive Industry Interaction Board (IIB) and a smaller External Advisory Board (EAB). The EAB, with which they interact closely, includes representatives from the utility sector, system vendors, and regulatory bodies, in addition to the DOE Office of Electricity Delivery and Energy Reliability and the DHS Office of Science and Technology.

## Partnerships and impact

While university led, TCIPG has always stressed real-world impact and industry partnerships. That is why TCIPG technologies have been adopted by the private sector.

- ▶ Several TCIPG technologies have been or are currently deployed on a pilot basis in real utility environments.
- ▶ A leading equipment vendor adopted their advanced technologies for securing embedded systems in grid controls.
- ▶ Three start-up companies in various stages of launch employ TCIPG foundational technologies.

To read more about TCIPG, visit <http://tcipg.org>. 

[Photo credit: SeanPavonePhoto/iStock/Thinkstock]



## References and further reading

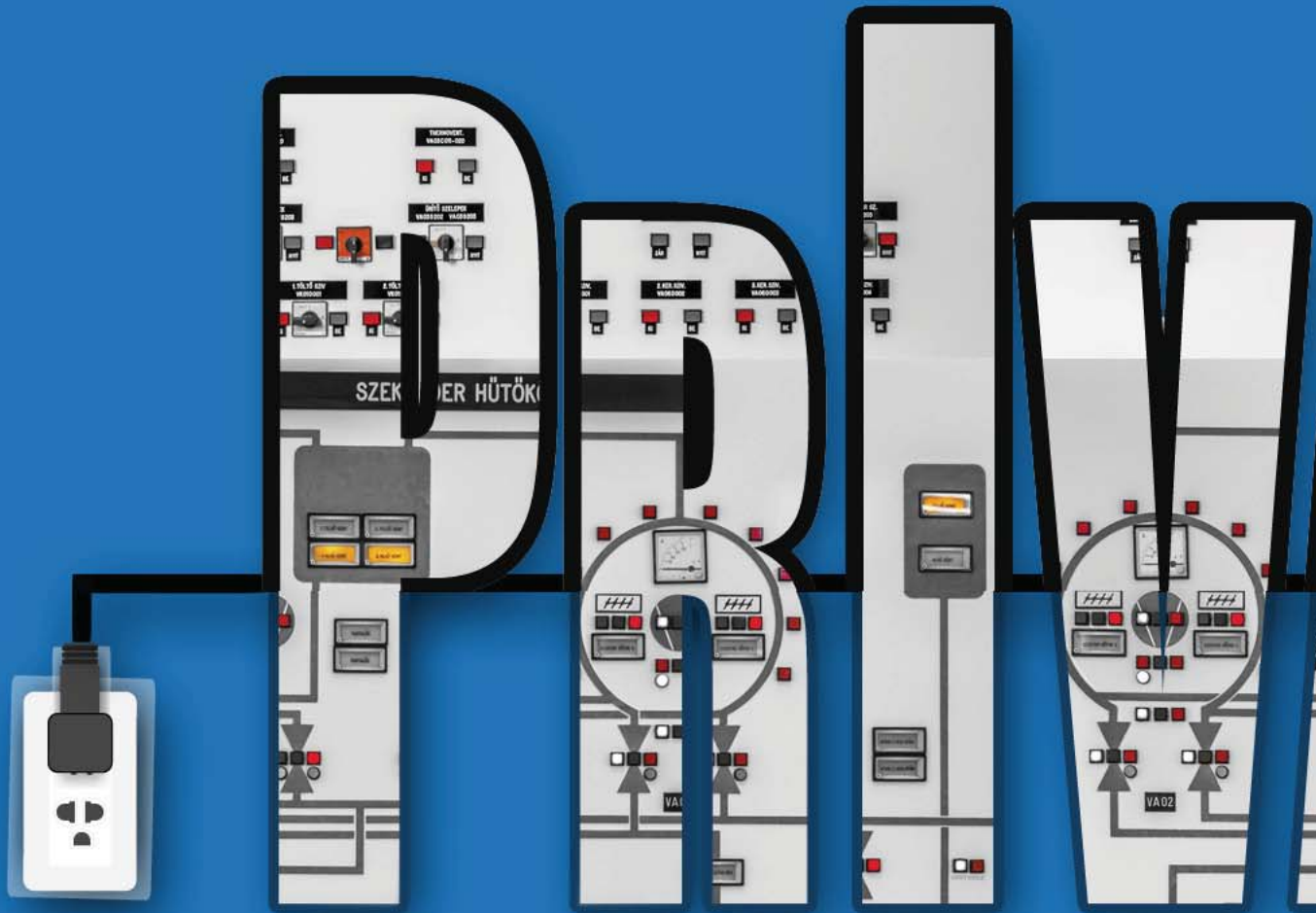
[1] *Homeland Security Presidential Directive 7: Critical Infrastructure Identification, Prioritization, and Protection*. December 17, 2003. Department of Homeland Security. Last Published Date: July 9, 2012. Available at: [www.dhs.gov/homeland-security-presidential-directive-7](http://www.dhs.gov/homeland-security-presidential-directive-7).

The following publications for further reading are available at <http://tcipg.org/research/>:

- ▶ CPINDEX: Cyber-physical vulnerability assessment for power-grid infrastructures
- ▶ Real time modeling and simulation of cyber-power system
- ▶ A hybrid network IDS for protective digital relays in the power transmission grid
- ▶ Power system analysis criteria-based computational efficiency enhancement for power flow and transient stability
- ▶ Cyber physical security for power grid protection
- ▶ An analysis of graphical authentication techniques for mobile platforms as alternatives to passwords
- ▶ Portunes: Privacy-preserving fast authentication for dynamic electric vehicle charging
- ▶ Secure data collection in constrained tree-based smart grid environments
- ▶ Practical and secure machine-to-machine data collection protocol in smart grid
- ▶ Searchable encrypted mobile meter
- ▶ Context-sensitive key management for smart grid telemetric devices



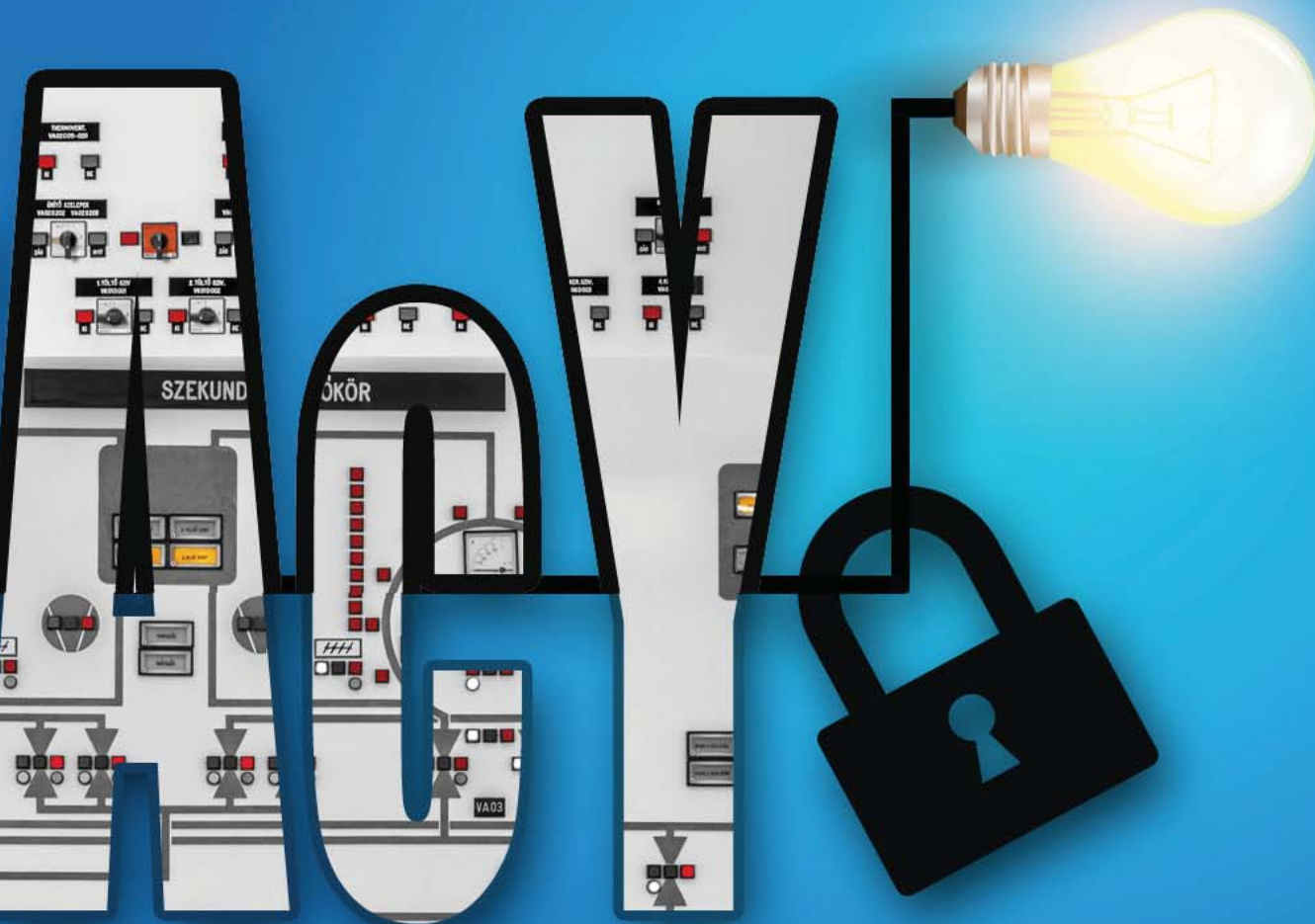




# Analyzing the cost of securing control systems\*

Zhenqi Huang | Yu Wang | Sayan Mitra | Geir Dullerud

\*This work is supported by NSA Science of Security grant (No. W911NSF-13-0086).



This article describes our recent progress on the development of rigorous analytical metrics for assessing the threat-performance trade-off in control systems. Computing systems that monitor and control physical processes are now pervasive, yet their security is frequently an afterthought rather than a first-order design consideration. We investigate a rational basis for deciding—at the design level—how much investment should be made to secure the system.

The level of investment to secure a control system typically depends on the particulars of the threats facing the system, as well as the perceived cost of a successful attack. Threat models are organized in terms of the attacker's skill, access, and available level of effort; the cost can be measured in terms of outage duration, number of users denied service, revenue lost, etc.

A few recent studies approach this problem as a game between the attacker and the defender in which the equilibrium point can inform the optimal investment in cybersecurity [1].

A more general approach takes the attacker out of the equation and looks at designs that are secure with provably low information leakage. Such systems are by design guaranteed not to leak any significant information to any external entity—be it an attacker or an ordinary bystander. By measuring the level of information leakage (e.g., how probable it is that an observer can correctly guess the private preferences of the users from observations), we can study the trade-off between the degree of leakage permitted and its cost on overall system performance.

## Distributed control with information sharing

We study this trade-off in a class of distributed control systems [2]. As one example, consider vehicles equipped with smart navigation devices that could share crowdsourced location and destination information to estimate traffic conditions more accurately [3]. Each vehicle's navigation device could then use aggregates of these estimates to select routes in light of current traffic conditions.

### Design scenarios

Multiple design possibilities or scenarios exist along the information-sharing continuum. At one extreme is absolute security, in which the vehicles' navigation devices never explicitly exchange information with each other. This architecture is absolutely secure as far as the communication channel goes; however, the vehicles miss out on the potential benefit of lowering their travel times using collaborative traffic estimation.

At the other extreme, the navigation devices communicate freely and share complete information. This

allows all of the devices to make accurate traffic predictions and optimal route choices but also increases the risk by making the devices vulnerable to a panoply of attacks that exploit the communication channels—distributed denial-of-service attacks, spoofing, violation of location privacy, etc.

Between these extremes lies a multitude of design possibilities that trade off security and performance differently. Similar trade-offs arise in other systems where users could sacrifice performance to achieve better security. In the power industry, for example, entities with different demand/consumption patterns could share (or not share) information in support of better demand estimates and better consumption scheduling.

## General framework

We present a general framework in which we model the traffic navigation scenario as:

$$\begin{aligned} u_i(t) &= g(p_i, x_i(t-1), \tilde{z}(t-1)) \\ x_i(t) &= f(x_i(t-1), u_i(t), z(t-1)) \\ \tilde{x}_i(t) &\sim r(x_i(t)) \\ z(t) &= h(x_1(t), \dots, x_N-1(t), z(t-1)) \\ \tilde{z}(t) &= h(\tilde{x}_1(t), \dots, \tilde{x}_{N-1}(t), \tilde{z}(t-1)) \end{aligned}$$

where  $u_i$  is the  $i^{\text{th}}$  agent's decision or control input at time  $t$ ; this is computed as a function of its preference  $p_i$  (say, destination), current state  $x_i(t-1)$  (position) and an estimate of the environment  $\tilde{z}(t-1)$  (traffic). This decision ( $u_i$ ) and the actual environment state determines its state in the next time step  $x_i(t)$ . The state that the  $i^{\text{th}}$  agent shares with the others is  $\tilde{x}_i(t)$ —this could be the exact value  $x_i(t)$  for the complete sharing strategy  $r_{cs}$ , or some randomized noisy version of it to give better privacy and/or security.

Finally,  $z(t)$  and  $\tilde{z}(t)$  are respectively the actual and estimated environment states computed as an aggregation ( $h$ ) of the actual and shared agent states. The cost of a particular communication strategy  $cost(r)$  is defined, for example, by summing up the distance to destination  $|x_i(t) - p_i(t)|$  of all agents. Then the cost of privacy is  $cost(r) - cost(r_{cs})$ , the cost of  $r$  relative to the cost of the complete information sharing strategy  $r_{cs}$ .

In [2], we specifically studied communication strategies that maintain differential privacy as introduced by Dwork et al. [4, 5]. In this case, we are concerned with privacy of the continuous sequence of locations



of the vehicles. We define a pair of agent preference sets  $p$  and  $p'$  to be *adjacent up to time  $T$*  provided they differ about the preference of at most one agent, say agent  $i$ , and that difference  $p_i - p'_i$  in the  $i^{\text{th}}$  agent's preferences is bounded.

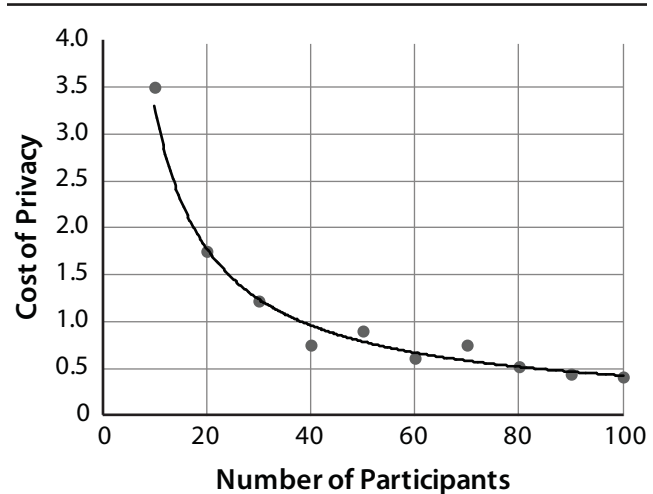
With this notion of adjacency, we say that a communication strategy maintains  $\epsilon$ -differential privacy (for small  $\epsilon$ ) under the following condition: the ratio of the probability that the communications came from a system with  $p$  to the probability that the communication came from system with preference vector  $p_i$  is at most  $e^\epsilon$ . That is, even an adversary with full access to all the communication in the system (the sequences  $\tilde{x}_i(t)$ ) can gain additional information about the preferences of an individual agent  $i$  with only negligible probability ( $e^\epsilon - 1$ ). Thus, the adversary is not able to make a high-confidence inference about any individual's preferences.

## Cost of privacy

Adapting the mechanisms in differential privacy literature to this setting, we first present a general communication mechanism for achieving  $\epsilon$ -differential privacy in distributed control systems. In this mechanism, each of the conveyed states is the actual state masked by a noise term that is chosen from a Laplacian distribution [ $p(x) = 1/2b e^{-|x|/b}$ , where the parameter  $b$  defines the variance of the distribution]. Specifically, in our mechanism, the parameter  $b$  is chosen as a function of  $\epsilon$  as well as the stability of the system.

With this design, we can precisely and mathematically characterize the trade-off between the level of security and privacy achieved and its cost. Specifically:

- ▶ We show that for a linear distributed system with quadratic cost functions, the standard deviation of the noise needed to make the system  $\epsilon$ -differentially private is independent of the size of the agent population. This is because increasing population has two opposing effects. On one hand, a larger number of agents are influenced by the changes in the preference of an individual agent  $i$ . On the other hand, the fractional influence of  $i$  on another individual agent through the environment weakens. In the linear case, these two effects roughly cancel each other. Since the means of Laplacian noise terms are zero, as the number of agents increases, the aggregate of noise terms converges to zero (see figure 1).



**FIGURE 1.** In this linear case, as the number of participants increases, the cost of privacy decreases.

- ▶ We also show that the required standard deviation of noise decreases with the stability of the dynamics and with the weakening of the environment's influence on an individual. Specifically, when the modulus of the maximum eigenvalue of the dynamics matrix is smaller, the effect of changes in an individual's preference on the system's trajectory decays faster over time. Consequently, for stable dynamics, if we fixed the total number of observations as the time horizon goes to infinity, the amount of noise required to achieve differential privacy becomes independent of the time horizon. For the unstable dynamics case, on the other hand, the amount of randomization needed can grow exponentially with the time horizon.

## Optimality of mechanisms

Adding the right amount of noise from the right distribution (Laplacian, in this case) gives  $\epsilon$ -differential privacy, and we can explicitly analyze the cost incurred. However, is this the most efficient (i.e., inexpensive) strategy for achieving  $\epsilon$ -differential privacy of control systems?

In [6], we answer this question by showing that indeed this Laplacian noise is the optimal choice for a subclass of the above control problem; namely, the class in which an individual agent aims to privately broadcast its states while minimizing the amount of noise added. The amount of noise is measured

by entropy in [6], and we can further show that the Laplacian noise is also the best choice to minimize a quadratic cost function of the agent at the same time. We show that, owing to the system's closed-loop nature, protecting the whole trajectory is equivalent to protecting the agent's preferences.

Therefore, the adversary is modeled by a filter estimating the system's initial state based on the system's randomized trajectory. We prove that if the system is  $\epsilon$ -differentially private to the adversary, then the entropy of the adversary's estimation has to be at least greater than a certain optimal value achieved via the Laplacian mechanism.

## Trade-offs in optimizing over a network

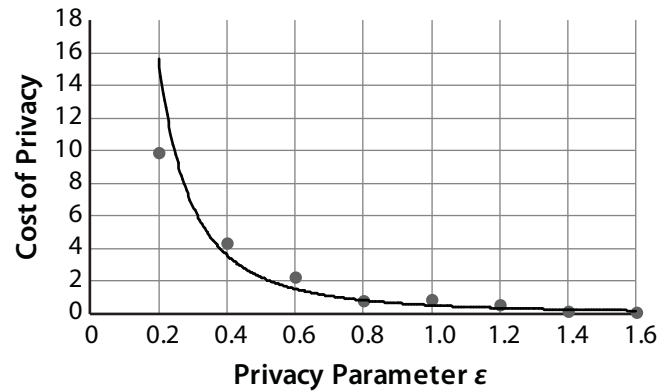
Another application of our framework is in understanding similar trade-offs in the context of distributed optimization problems [7]. Consider a scenario in which a collection of agents needs to select a common meeting point while minimizing the sum of the agents' individual travel costs. Each agent may have a different cost function and want to keep this cost function private (as cost functions can reveal an agent's priorities about time, distance, etc.).

In [7], we present mechanisms for solving such optimization problems while guaranteeing  $\epsilon$ -differential privacy of the cost functions. Our mechanism relies on all agents communicating noisy versions of their current guess about the meeting point, computing the average of these noisy guesses, and moving their guess towards a point that reduces their individual costs. The noise added to these guesses has to decay over the successive rounds.

With higher levels of privacy, shared information has to be noisier and the meeting point is more likely to be away from the optimal. We show that the expected deviation of the private consensus point, hence the cost or inaccuracy of the solution, from the true optimal point is of the order of  $O(1/\epsilon^2)$ . This is shown in figure 2.

## Conclusions


As a part of this project, we initiated the study of the trade-offs between performance or optimality of control systems and the level of communication security and privacy for which they could be designed.



**FIGURE 2.** As levels of privacy decrease (i.e., higher  $\epsilon$ ), the cost of privacy decreases. Each data point in this curve represents 500 numerical simulations, and the shape of the curve matches our analytical results.

We adopted the well-established notion of differential privacy in several of our papers because it is quantitative and could be extended naturally to continuously observed systems.

One practical implication of our analysis is that the proposed iterative, noise-adding mechanisms are more likely to be useful for stable systems with short-lived participants (e.g., drivers with short commutes). We also show that stable dynamics can work even with mechanisms independent of the number of participants.

Since the publication of our first result [8], several other connections have been drawn between differential privacy and control [9], filtering [10], and optimization [11]. Exploration of other security metrics, the corresponding performance trade-offs, optimality results, and their applications in realistic case studies all merit further attention—we believe that they will provide a scientific basis for security investments. 

## About the authors

**Zhenqi Huang** is a PhD candidate in electrical and computer engineering at University of Illinois at Urbana-Champaign (UIUC). He received his MS from UIUC in 2013. His research focuses on verification and synthesis for safe and secure controllers of cyber-physical systems.

**Yu Wang** is a graduate student of mechanical engineering at UIUC. He received his MS from UIUC

in 2014 and BS from Tsinghua University in 2012. His current research interest is the security of distributed and cyber-physical systems.

**Sayan Mitra** is an associate professor of electrical and computer engineering at the UIUC. He received his PhD from Massachusetts Institute of Technology in 2007. He held a postdoctoral fellowship at the California Institute of Technology and visiting positions at Oxford University and the Kirtland Air Force Research Laboratory in New Mexico. His research aims to develop algorithmic and software tools for design and analysis of distributed and cyber-physical systems. He received the National Science Foundation's CAREER Award, Air Force Office of Scientific Research Young Investigator Award, IEEE-HKN C. Holmes MacDonalld Outstanding Teaching Award, and several awards for best research paper.

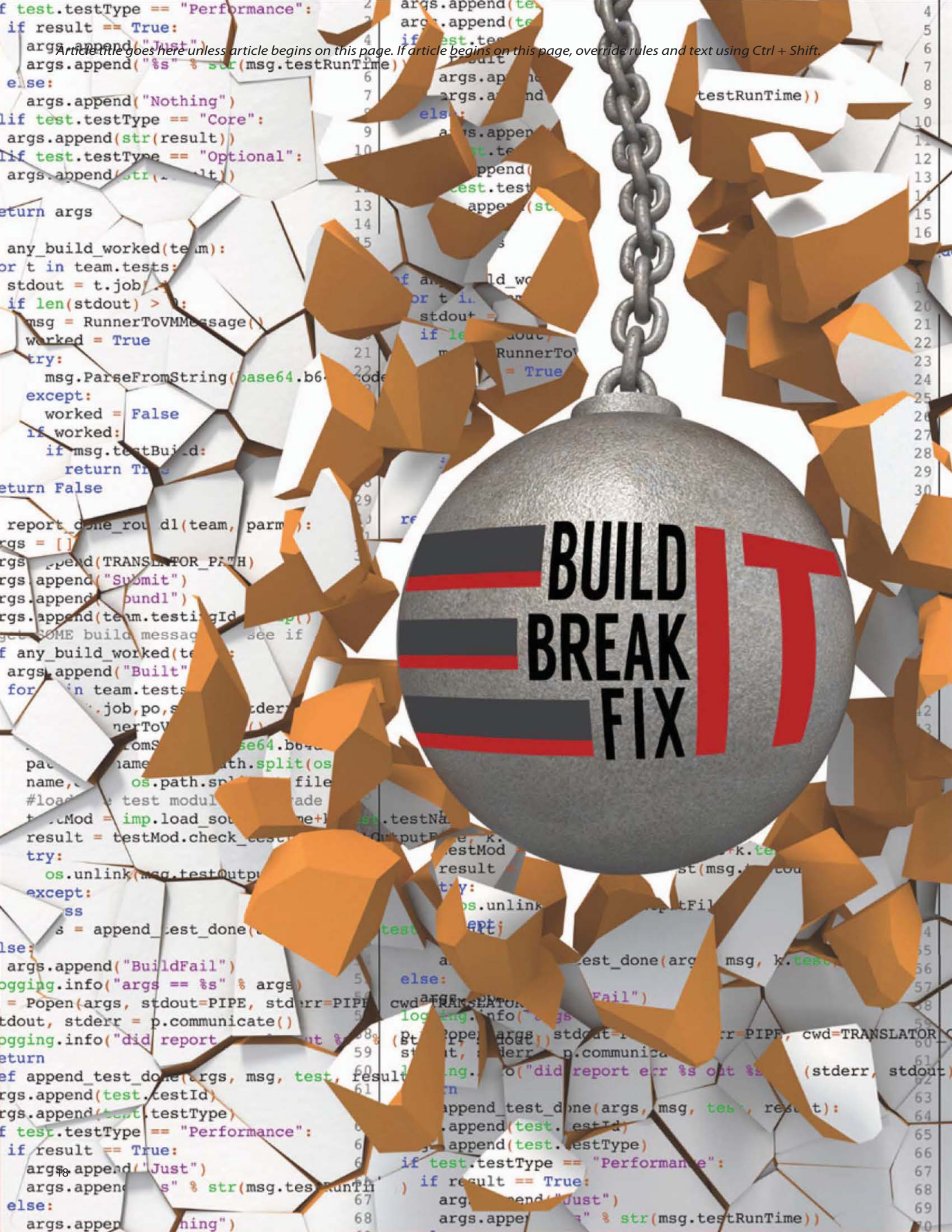
**Geir Dullerud** is a professor of mechanical science and engineering at UIUC, where he is the director of the Decision and Control Laboratory in the Coordinated Science Laboratory. He has held visiting positions in electrical engineering at KTH

(The Royal Institute of Technology), Stockholm, Sweden, in 2013, and in aeronautics and astronautics at Stanford University from 2005 through 2006. From 1996 to 1998, he was an assistant professor in applied mathematics at the University of Waterloo, Ontario, Canada. He was a research fellow and lecturer in the Control and Dynamical Systems Department, California Institute of Technology, in 1994 and 1995. He has published two books: *A Course in Robust Control Theory* (with F. Paganini) and *Control of Uncertain Sampled-Data Systems*. His areas of current research interests include cyber-physical systems security, games and networked control, robotic vehicles, and hybrid dynamical systems. Dr. Dullerud is currently an associate editor of the Society for Industrial and Applied Mathematics' *Journal of Control and Optimization*; he previously served in similar roles for both *IEEE Transactions on Automatic Control* and *Automatica*. He received the National Science Foundation CAREER Award in 1999 and the Xerox Faculty Research Award at UIUC in 2005. He is a fellow of the IEEE (2008) and of the American Society of Mechanical Engineers (2011).

## References

- [1] Panaousis E, Fielder A, Malacaria P, Hankin C, Smeraldi F. "Cyber-security games and investments: A decision support approach." In: Poovendran R, Saad W, editors. *Decision and Game Theory for Security*. New York: Springer, 2014. p. 266–286.
- [2] Huang Z, Wang Y, Mitra S, Dullerud GE. "On the cost of differential privacy in distributed control systems." In: *Proceedings of the Third International Conference on High Confidence Networked Systems*. 2014;105–114. doi: 10.1145/2566468.2566474.
- [3] Herrera JC, Work DB, Herring R, Ban XJ, Jacobson Q, Bayen AM. "Evaluation of traffic data obtained via GPS-enabled mobile phones: The Mobile Century field experiment." *Transportation Research Part C: Emerging Technologies*. 2010;18(4):568–583. doi:10.1016/j.trc.2009.10.006.
- [4] Dwork C. "Differential privacy: A survey of results." *Theory and Applications of Models of Computation, Lecture Notes in Computer Science*. 2008;4978:1–19. doi: 10.1007/978-3-540-79228-4\_1.
- [5] Dwork C, Naor M, Pitassi T, Rothblum GN. "Differential privacy under continual observation." In: *STOC '10: Proceedings of the 42nd ACM Symposium on Theory of Computing*. 2010:715–724. doi: 10.1145/1806689.1806787.
- [6] Wang Y, Huang Z, Mitra S, Dullerud G. "Entropy-minimizing mechanism for differential privacy of discrete-time linear feedback systems." In: 2014 IEEE 53rd Annual Conference on Decision and Control. 2014;2130–2135. doi: 10.1109/CDC.2014.7039713.
- [7] Huang Z, Mitra S, Vaidya N. "Differentially private distributed optimization." In: *Proceedings of the 2015 International Conference on Distributed Computing and Networking*. 2015. doi: 10.1145/2684464.2684480.
- [8] Huang Z, Mitra S, Dullerud G. "Differentially private iterative synchronous consensus." In: *Proceedings of the 2012 ACM Workshop on Privacy in the Electronic Society*. 2012;81–90. doi: 10.1145/2381966.2381978.
- [9] Le Ny J and Pappas GJ. "Privacy-preserving release of aggregate dynamic models." In: *Proceedings of the Second ACM International Conference on High Confidence Networked Systems*. 2013;49–56. doi: 10.1145/2461446.2461454.
- [10] Le Ny J and Pappas GJ. "Differentially private filtering." *IEEE Transactions on Automatic Control*, 2014;59(2):341–354. doi: 10.1109/CDC.2012.6426355.
- [11] Han S, Topcu U, Pappas GJ. "Differentially private convex optimization with piecewise affine objectives." In: *Proceedings of IEEE 53rd Annual Conference on Decision and Control*. 2014. Available at: <http://arxiv.org/abs/1403.6135>.





Article title goes here unless article begins on this page. If article begins on this page, override rules and text using Ctrl + Shift.

```
test.testType == "Performance":
if result == True:
    args.append("Just")
    args.append("%s" % str(msg.testRunTime))
else:
    args.append("Nothing")
if test.testType == "Core":
    args.append(str(result))
if test.testType == "Optional":
    args.append(str(result))

return args

def any_build_worked(team):
    for t in team.tests:
        stdout = t.job.stdout
        if len(stdout) > 0:
            msg = RunnerToVMMMessage()
            worked = True
            try:
                msg.ParseFromString(base64.b64decode(stdout))
            except:
                worked = False
            if worked:
                if msg.testBuild:
                    return True
    return False

def report_done_round(team, parms):
    args = []
    args.append(TRANSLATOR_PATH)
    args.append("Submit")
    args.append("round")
    args.append(team.testingId)
    get SOME build message see if
    if any_build_worked(team):
        args.append("Built")
        for t in team.tests:
            job, port, stdout, stderr = t.job
            runnerToVM = RunnerToVMMMessage()
            runnerToVM.FromString(base64.b64encode(stdout))
            path = os.path.split(os.path.join(team.testingPath, "test", t.testName))
            #load the test module
            testMod = imp.load_source("testMod", path[0])
            result = testMod.check_test(testMod, testMod, testMod)
            try:
                os.unlink(msg.testOutputPath)
            except:
                pass
            args = append_test_done(args, msg, test, result)
        else:
            args.append("BuildFail")
            logging.info("args == %s" % args)
            p = Popen(args, stdout=PIPE, stderr=PIPE, cwd=TRANSLATOR_PATH)
            stdout, stderr = p.communicate()
            logging.info("did report err %s out %s" % (stderr, stdout))
        return
    def append_test_done(args, msg, test, result):
        args.append(test.testId)
        args.append(test.testType)
        if test.testType == "Performance":
            if result == True:
                args.append("Just")
                args.append("%s" % str(msg.testRunTime))
            else:
                args.append("Nothing")
        else:
            args.append(str(result))
        if test.testType == "Performance":
            if result == True:
                args.append("Just")
                args.append("%s" % str(msg.testRunTime))
            else:
                args.append("Nothing")
        return args
```



```
if test.testType == "Performance":
    if result == True:
        args.append("Just")
        args.append("%s" % str(msg.testB
    else:
        args.append("Nothing")
elif test.testType == "Core":
    args.append(str(result))
elif test.testType == "Optional":
    args.append(str(result))

return args

def canBuildWorked(team):
    for test in team.test:
        stdout = t.join(
            if len(stdout) > 0:
                msg = RunnerToVMMMessage()
                worked = True
                t =
                msg.ParseFromString(base64.b64
            except:
                worked = False
            if worked:
                if msg.testBuild:
                    return True
    return False

def reportDoneRound(team, parms):
    = []
    .append(TRANSLATOR_PATH)
    s.append("Submit")
    id("Round")
    id(team.testingId.strip())
    build message to see if w
    d_worked(team):
        id("Build")

    for k in tests:
        tea, po, stdout, stderr,
        msg = k.toVMMMessage()
        sg.ParseFromString(base64.b64de
        path, filename = os.path.split(
        name, ext = os.path.splitext(file
        te = load the test module and grade
        testMod imp.load_source(name,
        msg.testReturnCode)
        testFile)
        try:
            outHalf, msg.testStdOut, msg.test
            os.unlink(msg.testOutputfile)
        except:
            pass
        args = append_test_done(args, msg
    else:
        args.append("BuildFail")
    logging.info("args == %s" % args)
    p = Popen(args, stdout=PIPE, stderr=
    stdout, stderr = p.communicate()
    logging.info("did report err %s out
    retur

def append_test_done(args, msg, test
args.append(test.testId)
args.append(test.testType)
if test.testType == "Performance":
    if result == True:
        args.append("Just")
        args.append("%s" % str(msg.testB
    else:
        args.append("Nothing")
```

# Build it, break it, fix it: Competing to build secure systems

Michael Hicks and Andrew Ruef

**W**e have a long legacy of failing to build secure systems; can a coding competition give us insight into what we can do better?

The news is filled with discussion of software that is supposed to be secure but is not. Do you ever wonder: **Why can't we get this right?** Why do people keep building software with the same kinds of defects, especially when there are known techniques for preventing them?

[Photo credit: mipan/iStock/Thinkstock]

## A change in mentality

Experts have long advocated that achieving security in a computer system requires careful design and implementation from the ground up. Researchers and technologists have sought to help by developing advanced analysis tools, testing methods, and programming languages and frameworks. Despite this increased attention, many developers seem oblivious to the experts' advice and continue to produce insecure software whose vulnerabilities could easily have been avoided. Several factors may be at work, including the development community's focus on finding problems in existing software—as exemplified by Capture-the-Flag (CTF) competitions—rather than on the best ways to build secure software.

We conceived Build-It, Break-It, Fix-It (BIBIFI) as a way to remedy this state of affairs in two ways. First, it aims to promote—to students and the wider community—a mentality of **building security in** rather than adding it after the fact. Second, BIBIFI aims to gather **evidence** about what works and what does not: By being open ended, the contest outcomes demonstrate effective methods for building secure software and breaking insecure software.

This article describes BIBIFI's three-round design involving both programming and exploiting, and our experience with the contest's first run in the Fall of 2014.

## Contest overview

Participants in the BIBIFI contest make up two sets of teams—the builder teams and the breaker teams—which are scored independently. The contest has three rounds: 1) build it, 2) break it, and 3) fix it. Builder teams participate in the build-it and fix-it rounds, while breaker teams participate in only the break-it round.

### Contest round 1: Build it

In BIBIFI's first, or build-it, round, we ask contestants to build a moderately complex software system that might be placed in security-relevant situations—for example, a web server or a file parser. Contestants can use whatever programming language or technology platform they wish, as long as it can be deployed to

the testing infrastructure we (i.e., the contest judges) use to judge it. We judge each submission by a pair of objective tests: 1) Does it meet predefined functionality requirements, and 2) what resources does it use when it runs? Submissions that pass a core set of correctness and performance tests are accepted and awarded points commensurate with their level of performance (i.e., the more efficient the correct software is, the more points it receives). Contestants may also implement optional features for additional points. The build-it round should provoke contestants to make observable trade-offs in development (e.g., between security and performance).

For the first contest run, we chose to have participants implement a secure log to describe activity in an art gallery, in particular when guests and employees enter and leave the gallery's rooms. The log must be stored in a single file, but no requirement is made on its format. The log is used by two programs. One program, *logappend*, appends new information to the log file. The other, *logread*, reads the log file to display the state of the art gallery according to a given query. Both programs use an authentication token, supplied as a command-line argument that must match the authentication token with which the log was created.

This contest problem has some basic security requirements. First, the implementation must ensure confidentiality; that is, without the token, an adversary should learn nothing about the contents of the log, even if he has direct access to the file. Second, it should ensure integrity; that is, without the token, an adversary should not be able to corrupt the file in a way that convinces *logread* or *logappend* (when using the correct token) that the log is reasonable.

### Contest round 2: Break it

In the second, or break-it, round, contestants perform vulnerability analysis on the software (including its source code) submitted during the first round. Round 2 contestants (who may or may not overlap with round 1 contestants) get points by finding vulnerabilities and other defects in the round 1 submissions, and those submissions with errors lose points. The break-it round fills the role of penetration testing by a red team and resembles other security contests, like DefCon CTF, whose focus is on finding vulnerabilities. In most contests, bugs are synthetic, introduced by the contest



organizers into a software package. In our contest, bugs are “real” because they are introduced by other contestants.

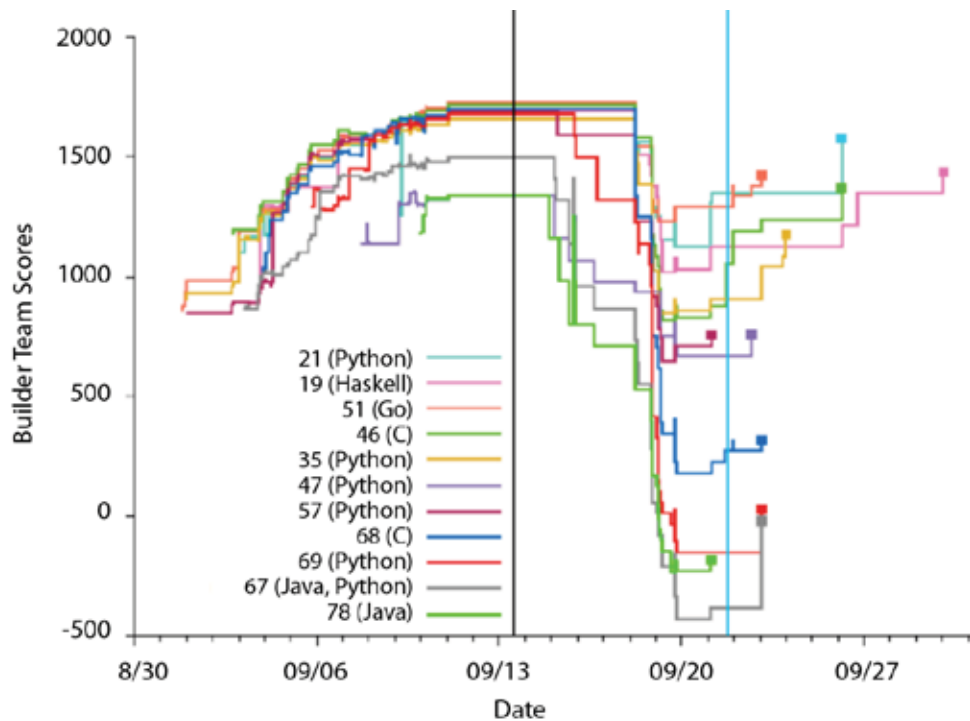
During the break-it round of the first contest run, breaker teams submitted security-related test cases that demonstrated problems. Teams could propose a violation of confidentiality by proving they knew something, despite the lack of an authentication token, about the hidden contents of a prepopulated log. To demonstrate an integrity violation, teams could submit a corrupted log (again, without knowledge of the authentication token) which *logread* and/or *logappend* nevertheless accepted as valid. Teams could also submit a working exploit (causing either program to core dump). Correctness bugs, demonstrated by submitted failing test cases, also were worth points (since they could ultimately be security related) but at only half the value of security bugs.

### Contest round 3: Fix it

In the final, or fix it, round, builder teams receive the test cases submitted by the breaker teams that identify failures in their code. For each test case, the team can submit a fix, and the fixed software is run against the remaining test cases, thereby potentially identifying many test cases that are “morally the same” (i.e., because the bug fix causes several test cases to pass). Judges manually confirm that a bug fix is really just for one bug, avoiding conflation of unrelated problems.

### Contest results

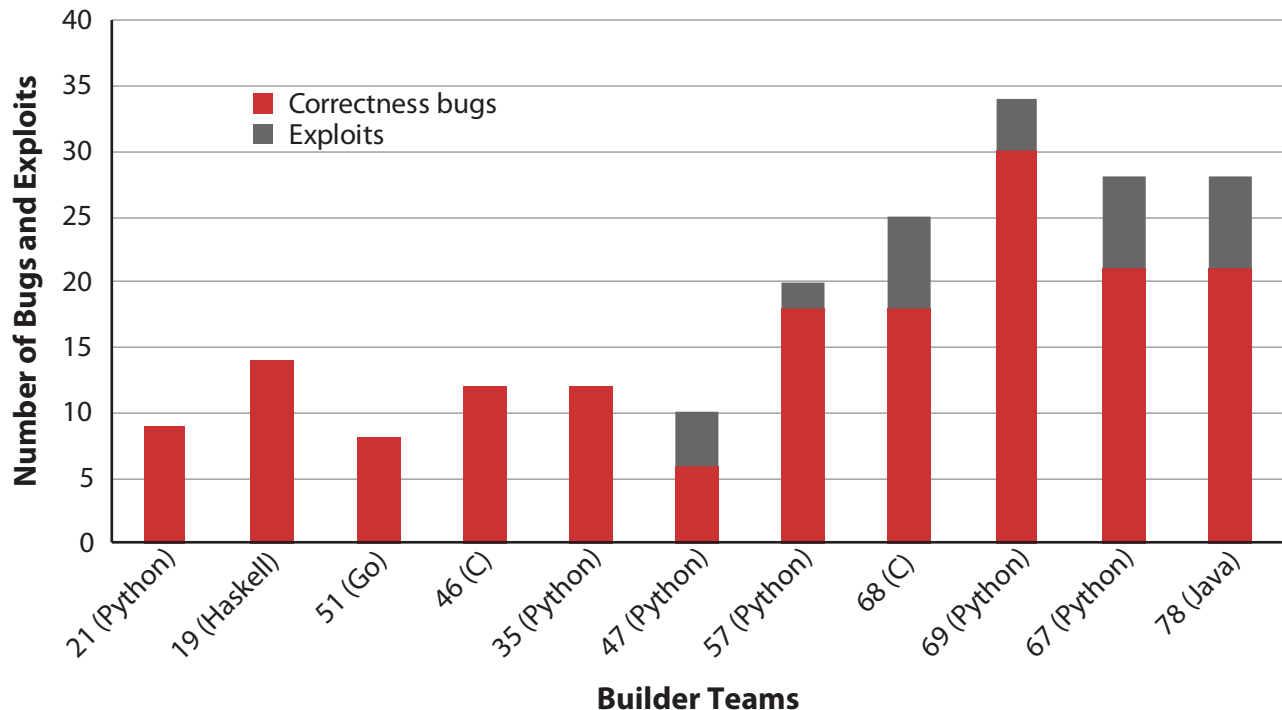
In figure 1, the x-axis of the chart is the date, with the starting point of the contest on the left, and the y-axis



**FIGURE 1.** The builder team scores over the course of the contest reveal a dramatic drop during the break-it round (marked by the black vertical line)—note that some teams finished with fewer than zero points. Teams at the bottom of the scoreboard did not correctly implement security features, and some did not even implement encryption or authentication. The key identifies each team by numeric identifier and programming language used.

is the score. Each line is one builder team, so the chart shows the score fluctuations over time as features are added and bugs discovered. The build-it round went from 8/28 to 9/9, while the break-it round went from 9/11 to 9/16. The drop in score during the break-it round is striking—note that some teams finished with fewer than zero points. Teams at the bottom of the scoreboard did not correctly implement security features, and some did not even implement encryption or authentication. It is interesting that of the two teams who wrote their programs in C, one had memory safety-related exploit issues, but the other did not. Figure 2 shows the number of correctness bugs versus exploits that breaker teams found in each builder teams’ submissions.


We can also study the activities and methodologies of participants in the break-it round. The winning breaker team (i.e., 51) was a very close third place builder team, and they reported that their bug identification method was driven by tests they prepared



**FIGURE 2.** The breaker teams found the following number of bugs and exploits in the builder teams' submissions. Builder teams appear on the x-axis in the order of their overall score (as indicated in figure 1). The red components of the bar are correctness bugs, while the grey components are exploits.

for themselves during their build-it activities. These results demonstrate that the capability to build secure systems can naturally translate into the capability to break systems.

## Next steps

We learned several lessons from running this contest, but ultimately we need more data to say anything conclusive. Our next scheduled contest began in early May 2015 as the Capstone of a Coursera online course specialization on cybersecurity. This run involved approximately 60 teams, rather than a couple of dozen, and we expect to extract a lot of compelling data, including the relative performance (i.e., security and efficiency) of programs written in different languages, and program metadata, such as a full list of changes made to the program over time. With this data we hope to gain new insights into what patterns in language design and software development methodologies improve security. If you have any questions you think we might try to answer with this contest or data, please contact us at [info@builditbreakit.org](mailto:info@builditbreakit.org). 

## Acknowledgements

This work was funded by the National Science Foundation, award 1319147. Professors Dave Levin, Atif Memon, and Jan Plane are coprincipal investigators with Michael Hicks. James Parker built the contest infrastructure and core web application (in Yesod for Haskell).

## About the authors

**Michael Hicks** is a professor in the Computer Science Department and the University of Maryland's Institute for Advanced Computer Studies (UMIACS) and is the former director of the Maryland Cybersecurity Center (MC2). His research focuses on using programming languages and analyses to improve the security, reliability, and availability of software. He is perhaps best known for his work exploring dynamic software updating, which is a technique of patching software without shutting it down. He has explored the design of new programming languages and analysis tools for helping programmers find bugs and





# The social engineering behind phishing



Christopher B. Mayhorn  
Emerson Murphy-Hill  
Olga A. Zielinska  
Allaire K. Welk

The social engineering behind phishing - Message (HTML)

Message Insert Options Format Text Developer Adobe PDF

Paste Calibri 11 A A Basic Text Address Book Check Names Attach File Attach Item Business Card Calendar Signature Follow Up Spelling

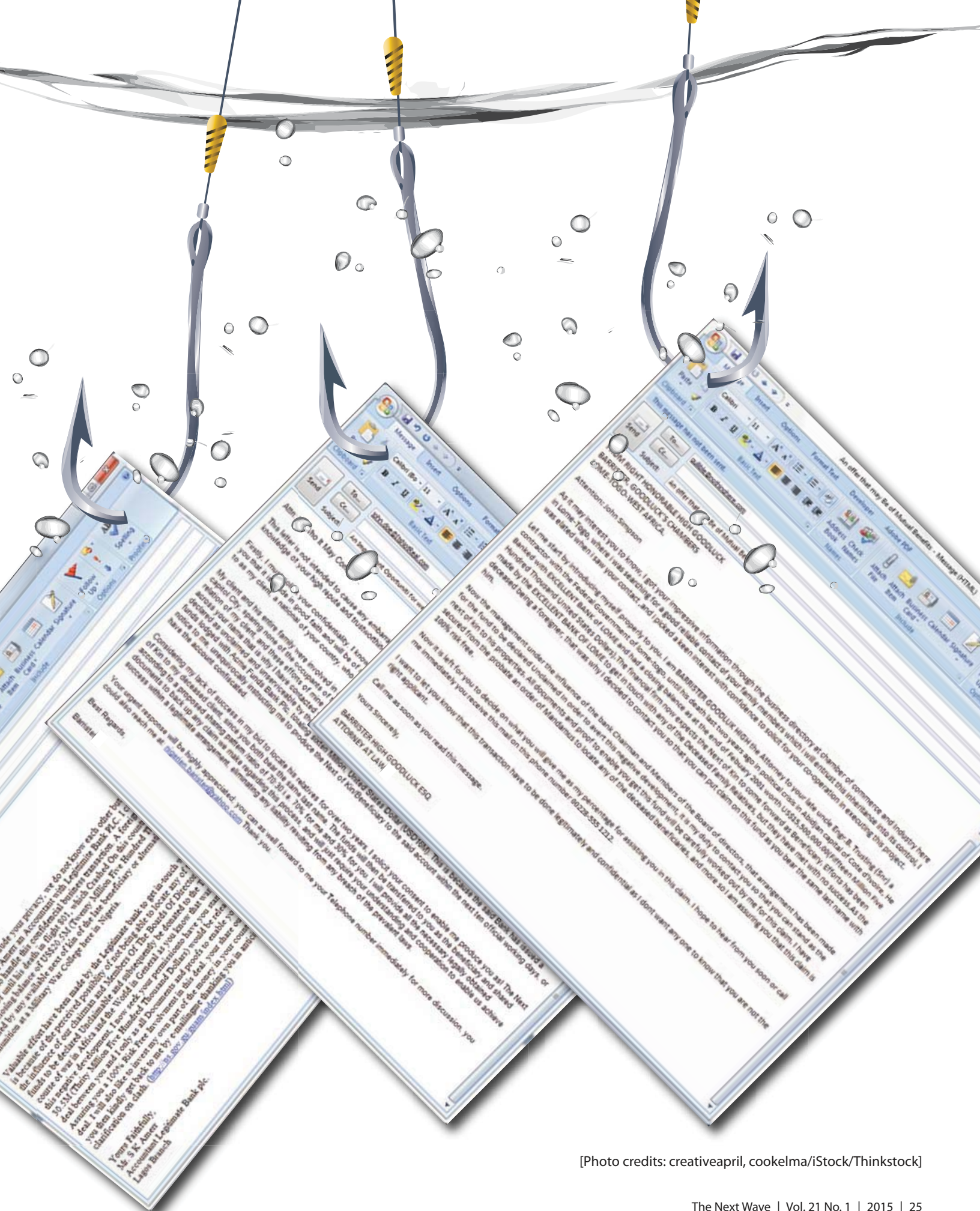
This message has not been sent.

Send To... TNW@tycho.ncsc.mil

Cc...

Subject: The social engineering behind phishing

**A**t one time or another, we have all probably received that suspicious e-mail from the English barrister informing us that a long-lost relative has passed away. Fortunately, this relative bequeathed his entire fortune to us, and all we have to do to receive this bounty is provide him with our bank account number! Most of us immediately recognize this communication as a *phishing* scam.



[Photo credits: creativeapril, cookelma/iStock/Thinkstock]



## Defining the phishing problem

Phishing is a social engineering tactic that cybercriminals use to trick people into revealing sensitive personal information, such as their date of birth, banking details, credit card information, or social security number. This is known as a *semantic* (i.e., language-based) attack because the criminals have targeted the computer user rather than technical aspects of the system. Users are typically sent an e-mail that appears to be from a known entity such as an established organization or individual that requires the user to reconfirm personal information by entering it via a supplied link within the text of the e-mail. These e-mails usually include “authentic” graphics and images that trick individual users into believing that the communication and request for information is legitimate.

If all attacks were so obvious, most people would have little trouble avoiding an adverse outcome. While communications ostensibly from Nigerian princes and former Iraqi generals are always suspect, criminal activities are becoming increasingly more frequent and difficult to detect. For instance, Kaspersky Lab reported that there were as many as 37.3 million attacks in 2013, up from 19.9 million in 2012 [1]. Given the sheer number of attacks, it is likely that a percentage will be successful. After all, criminals would not engage in this activity if some individuals did not provide the requested information. For the phishing victim, personal costs associated with falling prey to such an attack can include loss of time, increased stress, monetary losses, and damaged credit. Some estimates indicate that each individual phishing attack costs approximately \$866, and phishing attacks overall contribute to over \$3 billion in annual financial losses [2].

These direct costs of phishing to individuals are joined by other direct costs such as those incurred by private sector financial institutions as they attempt to shore up compromised systems and fix damaged credit. Likewise, less direct costs might be incurred by legitimate business entities that lose profits as users become more hesitant to trust online access. Costs continue to grow as government assets are deployed for investigation and enforcement purposes.

Faced with these disheartening statistics along with a steadily increasing price tag, what can be done to prevent people from being victimized? Previous efforts to combat the phishing problem have focused

on building technological solutions, such as phishing web-page detectors, yet some authors (e.g., [3]) have suggested that regardless of how security-related technology is improved, successful solutions must address the “people problem.” The purpose of this article is to describe and summarize our NSA-funded research program at North Carolina State University (NCSU) that attempts to address this topic.

The NCSU Science of Security Lablet, one of only four in the United States funded by the NSA, has begun to investigate phishing using a multidisciplinary approach. Team members come from diverse backgrounds, including the university’s departments of psychology and computer science.

Below, we describe a series of studies that sought to answer broad questions regarding who is at risk, what factors predict phishing susceptibility, and how phishing susceptibility might be reduced through the implementation of training programs. Lastly, we conclude with a section that describes how our findings might inform the design of future tools that implement tailored warning systems.

## Who is at risk?

To better understand who is at risk when confronted with a phishing e-mail, we conducted an initial survey that asked 155 respondents to describe their previous experiences with phishing attempts and the related consequences [4]. Virtually all participants indicated that they had received a phishing e-mail at some time in the past, and 22% reported that these attempts were successful. In addition, 84% of participants readily identified e-mail as the media where they were most likely to encounter phishing messages, but participants also described other instances where phishing messages were delivered via instant messaging, job boards, or social networking sites. As the following response indicates, phishers are becoming very creative in their efforts:

I applied for a part time job through Craigslist and had to do a credit check to successfully apply. I thought it was OK since lots of employers now do credit checks. I entered my social and lots of other information. . . . By next week I had several pings in my credit report of suspicious activity. Someone had taken out a credit card in my name and also tried to get



a loan. I was scared, honestly, that someone could use my information in that way. I was also angry . . .

When asked about the content of phishing messages, qualitative comments from respondents suggested that phishing communications often sound “too good to be true” and include “exciting or unbelievable offers.” In addition, comments also revealed phishing attacks often use a “strong pitch,” and attempt to elicit “a feeling of urgency to get stuff done now,” by using “a limited time offer or high-pressure tactics” in an attempt to get victims to act quickly.

Although we believed the costs of getting phished were obvious, these results are informative because they indicate that the effects are not limited to financial costs or loss of material items only (e.g., money, property, etc.), but may have social ramifications as well (e.g., loss of trust, embarrassment). Qualitative comments underscored potential psychological impacts resulting from phishing attacks; participants referenced negative emotions, such as “embarrassment, shame or loss of self-confidence.”

## What makes someone susceptible to phishing attacks?

Because we are all apparently at risk when it comes to phishing attempts, our next efforts were to clarify why users might be at risk. Previous research indicated that cognitive factors, such as attentional vigilance to cues in the computing environment, serve as a key component in avoiding phishing [5, 6]. Other studies have identified how users who fall prey to phishing tend to haphazardly rely on perceptual cues, such as the layout of a webpage, or on social cues, such as whether or not the sender of an e-mail is known [7]. In effect, users try to ascertain the veracity of cues to determine whether they can trust the sender prior to making a security-related decision. This is problematic because criminals often manipulate aspects of digital communications that cultivate trust, thereby increasing phishing susceptibility [8].

As people tend to vary with regard to individual differences in cognition, perception, and dispositional factors, we sought to determine what factors make some users more susceptible to phishing than others [9]. In this particular study, 53 undergraduate

students completed a battery of cognitive tests and a survey designed to assess impulsivity, trust, and personality traits before they performed an e-mail categorization task that required them to discriminate legitimate e-mails from phishing attempts.

Our results indicated that individuals who possessed personality characteristics such as reserved behavior consistent with introverts, low impulsivity, and decreased trust were more likely than others to accurately identify phishing messages. Likewise, previous experience such as suffering a monetary loss also decreased susceptibility to phishing attacks. These findings taken together suggest that some people are more susceptible to phishing attacks than others, so efforts to ameliorate phishing might work best if efforts are focused on those most at risk (i.e., those who are extroverted, impulsive, and trusting).

Because these are measurable characteristics and there are a variety of psychological instruments available to assess these behavioral constructs, it is feasible that a quantifiable profile of phishing susceptibility could be constructed. While promising, such efforts would need to be validated empirically and psychometrically.

Although the previous work suggests that individual differences are important determinants of phishing susceptibility, human behavior does not occur in a vacuum. One caveat that has pervaded social science research for the last 150 years is that behavior varies by social context. Given increasing workplace diversity and the globalization of the business industry coupled with enhanced communication enabled by technology, interaction with geographically distributed multinational teams is now commonplace to most of us.

Extending the concept of individual differences to group differences begs the question of whether culture plays a role in phishing susceptibility. To answer this question, we examined self-reported rates of phishing susceptibility and online privacy behaviors from Chinese, Indian, and American samples [10]. We surveyed 164 participants from the United States, India, and China to assess past phishing experiences and the likelihood of engaging in online safety practices (e.g., reading a privacy policy). Results indicated that all nationalities were equally likely to experience phishing attempts yet the prevalence of being successfully phished varied by nationality such that

only 9% of Chinese, 14% of Americans, and 31% of Indians had been successfully phished. Thus, Chinese and American respondents were about as likely to get phished yet both of these nationalities were less susceptible than Indian respondents.

We discussed these potential cultural differences in terms of power distance—where low power distance countries, such as the United States, could be considered individualistic and more challenging of authority than high power distance countries, like India, that tend to convey high levels of respect to authorities where compliance with information requests might be more likely.

With regard to taking protective action to prevent information loss, cultural differences were also observed such that Chinese and Americans were more likely than Indian respondents to report destroying old documents, yet Americans were more likely than either Chinese or Indians to actively search a web page for the secure padlock icon when making online transactions. These results suggest that cultural background might be another factor to consider when developing a profile of phishing susceptibility. Such a profile would theoretically be useful in identifying those most in need of security training.

## Can training prevent phishing?

Antiphishing training is one approach to making the user aware of phishing thereby acting as a barrier to attacks [11]. In the past, antiphishing training has ranged from a list of Internet user tips to a cartoon that helps explain user tips in a story format to even a game that provides embedded training against phishing [12]. From past research, training efforts were more effective when shown in a real-world context [13]. Additionally, another study revealed that the level of threat perception determines the quality of protective action taken because perception of a high level of threat motivated participants to act and change their behavior. Likewise, such threat manipulations also increased the retention of information [14].

Given these general considerations regarding the development of an antiphishing training program, we developed two experimental antiphishing training conditions: one that conveyed real-world consequences to trainees, and one that attempted to induce perceptions of high threat [15]. The training on real-world consequences was delivered via three videos

that reported on different news stories where identity theft occurred as a result of phishing, followed by an emotional interview with a victim of a fake money order scam. The second training condition used three news articles selected with the intention of raising the level of threat perceived by participants. These articles included recent news stories about how Facebook is collecting data and selling it along with news stories regarding the recent leak at NSA perpetrated by an insider. These two experimental antiphishing training conditions were compared to a third control condition that showed participants a cooking video.

Ninety-six participants completed a baseline e-mail categorization task in which they had to discriminate legitimate e-mails from phishing attempts before being randomly assigned to one of the three training conditions. After training was completed, a second e-mail categorization task was completed. An increased rate of accurately identifying phishing e-mails on the second task compared to the baseline was observed in all training conditions—suggesting that training was generally effective. Unfortunately, there were no statistically significant differences between the experimental training conditions and the control condition; although, trends suggested that heightening the threat perception slightly enhanced participants' abilities to detect phishing messages.

While these particular training manipulations did not produce compelling results, another approach would be to train individuals less experienced with computer security on how experts conceptualize phishing attacks. In essence, such training would allow novices to learn from more experienced experts.

## How novices and experts conceptualize attacks

One method to quantify differences in experience includes examining differences between the mental models of security novices and experts. Mental models are internal representations that users develop of a concept or system. Mental models grow as individuals interact with a system or concept; eventually, the user will be able to use his or her developed mental models to predict or explain the system or concept [16]. Accordingly, as users develop expertise, they have qualitative changes in their mental models. Experts are able to quickly analyze a situation or case and make

quick decisions because of their coherent organization of information. Thus, an underlying tenet of naturalistic decision-making research [17] suggests that training novices to use expert-level tactics might be useful in reducing errors (in this case, reducing phishing susceptibility).

Our most recent phishing-related project assessed how the mental models of computer security novices varied from those of computer security experts [18]. Twenty-eight participants (20 novices and 8 experts) were asked to rate the strength of the relationship among pairs of phishing-related concepts. These relatedness ratings were entered into Pathfinder, a statistical software tool that represents pairwise proximities in a network [19]. Preliminary findings suggest that novices and experts had significantly different mental models with regard to the prevention of phishing attacks and the trends and characteristics of attacks.

Expert mental models were more complex with more links between concepts, and this could have implications for training. For example, the aggregate expert model illustrated “unknown sender” as a central node connected to “social engineering,” “legitimate appearance,” “link,” “attachment,” and “bad spelling/grammar”; whereas, novices only linked “unknown senders” to “attachment” and “link.” This illustrates that experts likely have a more comprehensive understanding of how unknown senders can relate to a broad array of phishing trends and characteristics. Training programs might aim to replicate this expert model in novices by providing information regarding the interconnectedness of these trends and characteristics related to unknown senders.

## Future directions


While efforts to promote cybersecurity through training might yet prove to be an effective means to reducing phishing susceptibility, it is unclear whether users will be motivated to spend the time and energy to attend such sessions. Also, it is unrealistic to presume that people will be constantly on guard to protect themselves from potential online security threats, so perhaps this function should be allocated to the technology involved. It is likely that such a technology would include some type of warning functionality that would serve to alert users when their information is at risk. To address the potential characteristics of such a

system, there are a number of theoretical frameworks within the hazard communication literature that have been used to describe response to warning messages where some action has to be taken when a threat is detected [20, 21, 22].

In all of these theoretical models, members of the public encounter a warning message that describes the nature of a hazard and suggests courses of action to avoid the consequences. Ultimately, the individual decision maker must act to either comply with or ignore the warning message. A growing realization within the hazard communication literature is that effective warning messages must be tailored to match the hazardousness of the situation or to the user’s characteristics to benefit comprehension [23]. Our initial efforts described above provide data to build a profile of at-risk users who are especially susceptible to phishing thereby providing the knowledge necessary to tailor effective warning messages. For instance, foreknowledge of a user’s impulsive nature from previous online activities might suggest that the inclusion of an “Are you sure?” dialog box following an initial attempt to follow a suspicious link might result in temporal delay that allows a more thoughtful response. However, this example also illustrates that the development of such a tool must include a consideration of usability and technology adoption to ensure that potential solutions are acceptable to users [24].

## Conclusions

Given the potential costs to individuals, organizations, and governments, phishing is a cybersecurity problem that demands attention in terms of both research and practice. As the results described above indicate, we are starting to answer some important questions that can be useful in designing countermeasures to reduce the likelihood of data loss. By understanding how individual differences in cognition, perception, and behavior predict phishing susceptibility, we can identify and target vulnerability for training interventions. We have already investigated whether or not specific training tactics help to reduce phishing susceptibility, but much more work needs to be done.

Lastly, we have begun to compile a set of functional requirements to guide development of future technological tools that help to protect our information in cyberspace. 



## About the authors

**Christopher B. Mayhorn** is a professor and the program coordinator of the Human Factors and Applied Cognition Psychology Program at North Carolina State University (NCSU). He earned a BA from The Citadel in 1992, an MS in 1995, a Graduate Certificate in Gerontology in 1995, and a PhD in 1999 from the University of Georgia. He also completed a postdoctoral fellowship at the Georgia Institute of Technology. Dr. Mayhorn's current research interests include everyday memory, decision making, human-computer interaction, and safety and risk communication. Dr. Mayhorn has more than 50 peer-reviewed publications to his credit, and his research has been funded by government agencies such as the National Science Foundation and the NSA. Currently, he serves as the chair of the Technical Program Committee for the Annual Meeting of the Human Factors and Ergonomics Society and is on the editorial board of *Human Factors*. He was also recently named one of eighteen University Faculty Scholars at NCSU.

**Emerson Murphy-Hill** is an assistant professor in computer science at NCSU. He earned a BS from The Evergreen State College in 2004 and a PhD from Portland State University in 2009. He also completed a postdoctoral fellowship at the University of British Columbia in 2010. He directs the Developer Liberation Front, a research group dedicated to exploring the intersections of human-computer interaction and software engineering.

**Olga A. Zielinska** is a PhD student in the Human Factors and Applied Cognition Psychology Program at NCSU. She recently earned her MS degree from NCSU; her thesis focused on how the formatting of search results can affect how Internet users search for health information. She has a BS in biomedical engineering from Drexel University.

**Allaire K. Welk** is a PhD student in the Human Factors and Applied Cognition Psychology program at NCSU. She recently earned her MS degree from NCSU; her thesis focused on the phenomenon of inattention blindness as it relates to informational displays. She has a BA in psychology from NCSU.

## References

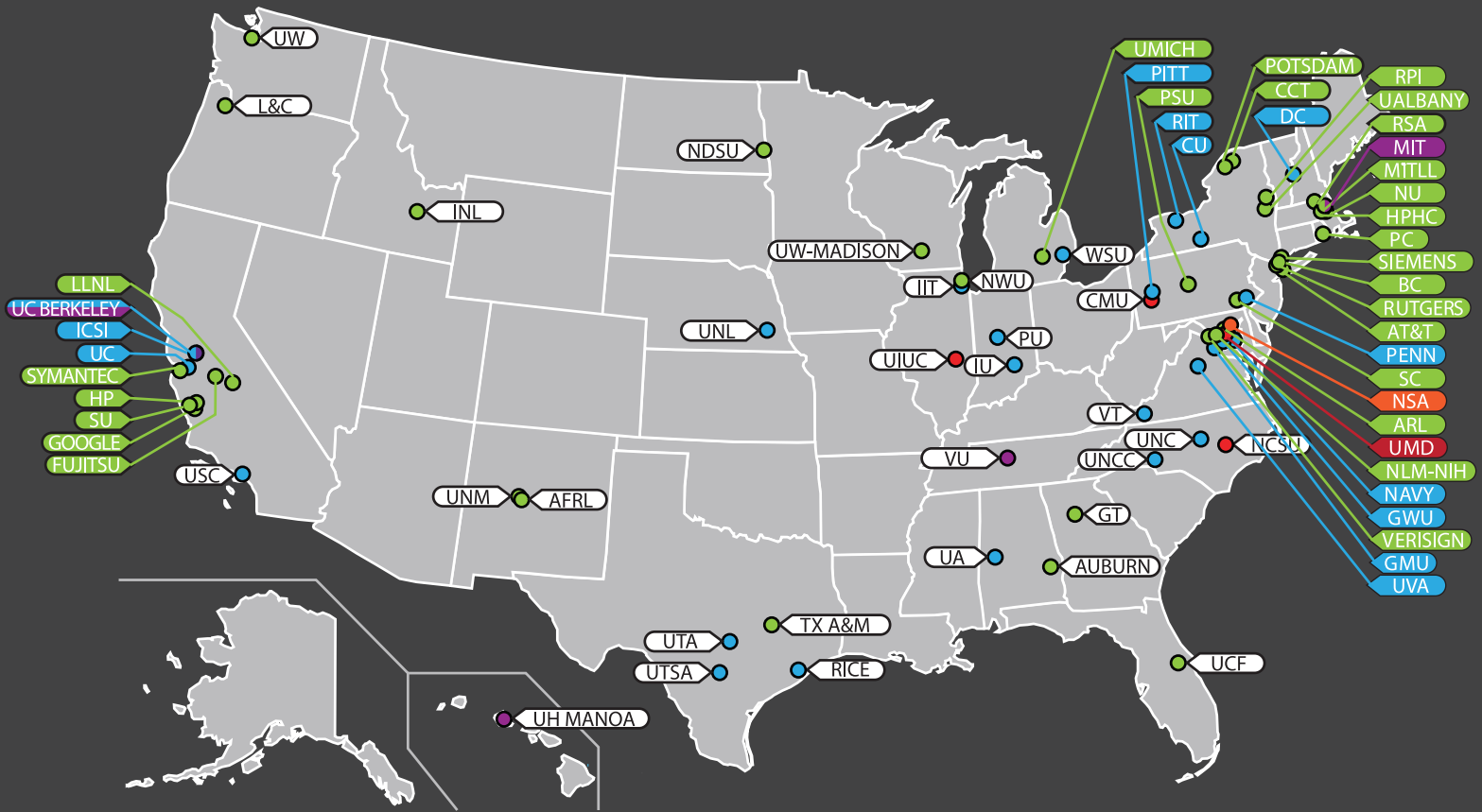
- [1] Kaspersky Lab. "37.3 Million Users Experienced Phishing Attacks in the Last Year" [Press Release]. 20 Jun 2013. Available at: [http://www.kaspersky.com/about/news/press/2013/Kaspersky\\_Lab\\_report\\_37\\_3\\_million\\_users\\_experienced\\_phishing\\_attacks\\_in\\_the\\_last\\_year](http://www.kaspersky.com/about/news/press/2013/Kaspersky_Lab_report_37_3_million_users_experienced_phishing_attacks_in_the_last_year).
- [2] Gartner. "Gartner survey shows phishing attacks escalated in 2007; more than \$3 billion lost to these attacks" [Press release]. 17 Dec 2007. Available at: <http://www.gartner.com/it/page.jsp?id=565125>.
- [3] Schneier B. *Secrets and Lies: Digital Security in a Networked World*. New York (NY): Wiley & Sons; 2000. ISBN-13: 978-0-471-45380-2.
- [4] Kelley CM, Hong KW, Mayhorn CB, Murphy-Hill E. "Something smells phishy: Exploring definitions, consequences, and reactions to phishing." In: *Proceedings of the Human Factors and Ergonomics Society 56th Annual Meeting*; 2012. doi: 10.1177/1071181312561447.
- [5] Downs JS, Holbrook MB, Cranor LF. "Decision strategies and susceptibility to phishing." In: *Proceedings of the Second Symposium on Usable Privacy and Security*; 2006. doi: 10.1145/1143120.1143131.
- [6] Vishwanath A, Herath T, Chen R, Wang J, Rao HR. "Why do people get phished? Testing individual differences in phishing vulnerability with an integrated, information processing model." *Decision Support Systems*. 2011;51(3):576–586. doi: 10.1016/j.dss.2011.03.002.
- [7] Jagatic TN, Johnson NA, Jakobsson M, Menczer F. "Social phishing." *Communications of the ACM*. 2007;50(10):94–100. doi: 10.1145/1290958.1290968.
- [8] Workman M. "Wisecrackers: A theory-grounded investigation of phishing and pretext social engineering threats to information security." *Journal of the American Society for Information Science and Technology*. 2008;59(4):662–674. doi: 10.1002/asi.v59:4.

- [9] Mayhorn CB, Welk AK, Zielinska OA, Murphy-Hill E. "Assessing individual differences in a phishing detection task." In: *Proceedings of the 19<sup>th</sup> World Congress of the International Ergonomics Association*. Melbourne, Australia. (In press).
- [10] Tembe R, Zielinska O, Liu Y, Hong KW, Murphy-Hill E, Mayhorn CB, Ge X. "Phishing in international waters: Exploring cross-cultural differences in phishing conceptualizations between Chinese, Indian, and American samples." In: *Proceedings of HotSoS: Symposium and Bootcamp on the Science of Security*; 2014. doi: 10.1145/2600176.2600178.
- [11] Mayhorn CB, Nyeste PG. "Training users to counteract phishing." *Work*. 2012;41(Suppl 1):3549–3552. doi: 10.3233/WOR-2012-1054-3549.
- [12] Alnajim A, Munro M. "An anti-phishing approach that uses training intervention for phishing websites detection." In: *Sixth International Conference on Information Technology: New Generations, 2009. ITNG '09*; 2009. doi: 10.1109/ITNG.2009.109.
- [13] Kumaraguru P, Sheng S, Acquist A, Cranor L, Hong J. "Lessons learned from a real-world evaluation of anti-phishing training." In: *eCrime Researchers Summit*; 2008. doi: 10.1109/ECRIME.2008.4696970.
- [14] Davinson N, Sillence E. "It won't happen to me: Promoting secure behaviour among internet users." *Computers in Human Behavior*. 2010;26(6):1739–1747. doi: 10.1016/j.chb.2010.06.023.
- [15] Zielinska OA, Tembe R, Hong KW, Xe G, Murphy-Hill E, Mayhorn CB. "One phish, two phish, how to avoid the Internet phish: Analysis of training strategies to detect phishing e-mails." In: *Proceedings of the Human Factors and Ergonomics Society 58th Annual Meeting*; 2014. doi: 10.1177/1541931214581306.
- [16] Norman DA. "Some observations on mental models." In: Genter D, Stevens AL, editors. *Mental Models*. Hillsdale (NJ): Erlbaum; 1983. p. 7–14.
- [17] Klein G. *Sources of Power*. Cambridge (MA): MIT Press; 1999. ISBN-13: 978-0262112277.
- [18] Zielinska OA, Welk AK, Murphy-Hill E, Mayhorn CB. (in press). "Exploring expert and novice mental models of phishing." In: *Proceedings of HotSoS: Symposium and Bootcamp on the Science of Security*; 2015. doi: 10.1145/2746194.2746216.
- [19] Rowe AL, Cooke NJ. "Measuring mental models: Choosing the right tools for the job." *Human Resource Development Quarterly*. 1995;6(3):243–255. doi: 10.1002/hrdq.3920060303.
- [20] Lindell MK, Perry RW. (2004). *Communicating Environmental Risk in Multiethnic Communities*. Thousand Oaks (CA): Sage Publications; 2004. ISBN: 0-7619-0650-9.
- [21] Rogers WA, Lamson N, Rousseau GK. "Warning research: An integrative perspective." *Human Factors*, 2000;42(1):102–139. doi: 10.1518/001872000779656624.
- [22] Wogalter MS, Dejoy, DM, Laughery, KR. *Warnings and risk communication*. London: Taylor and Francis; 1999. ISBN: 0748402667.
- [23] Wogalter MS, Mayhorn CB. "Providing cognitive support with technology-based warning systems." *Ergonomics*, 2005;48(5):522–533. doi: 10.1080/00140130400029258.
- [24] Venkatesh V, Davis F D. "A theoretical extension of the technology acceptance model: Four longitudinal field studies." *Management Science*. 2000;46(2):186–204. doi: 10.1287/mnsc.46.2.186.11926.



# GLOBE AT A GLANCE

## NSA Science of Security Research Network

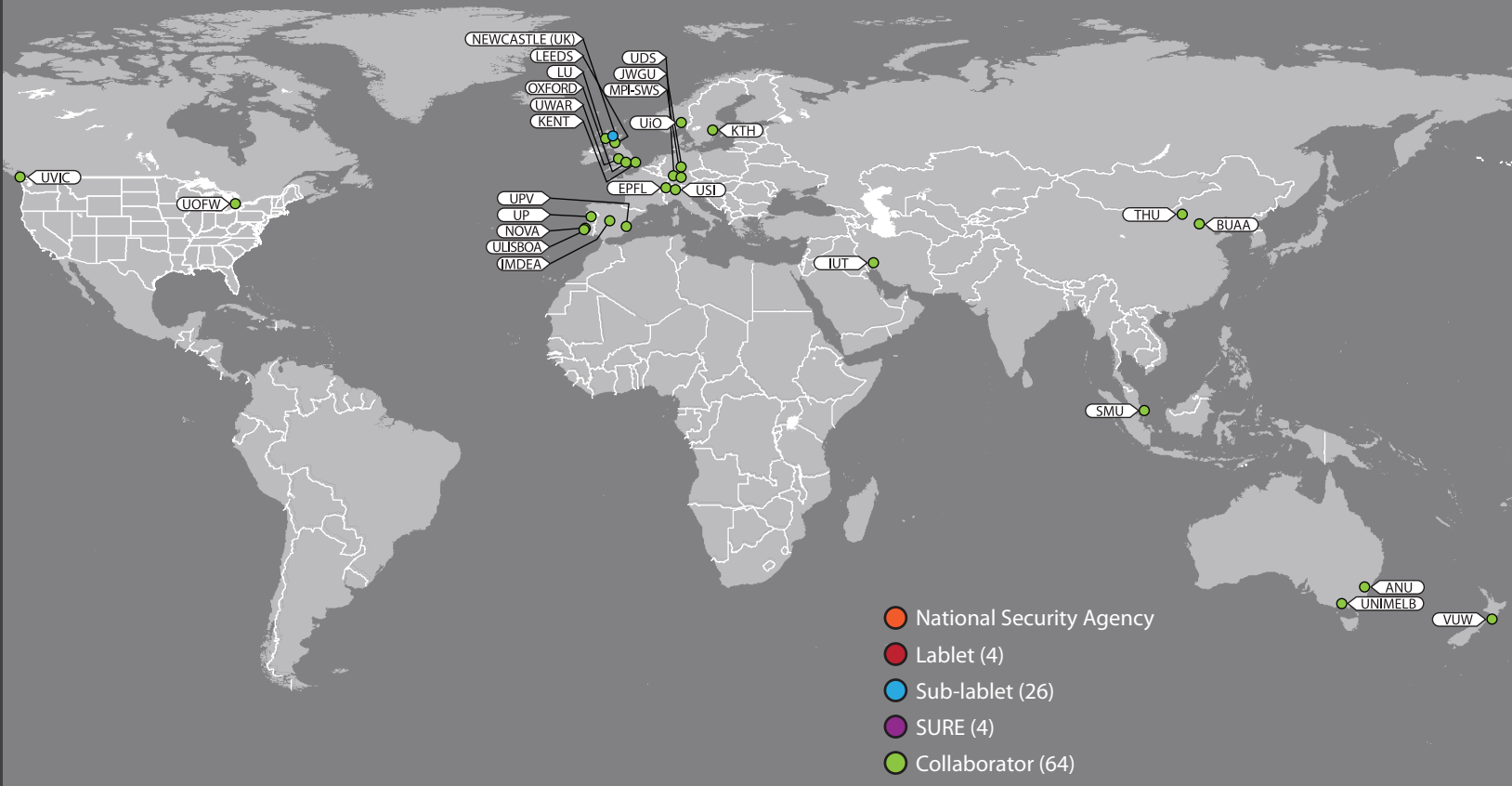


### ABBREVIATIONS

- |   |   |   |   |
|---|---|---|---|
| ● AFRL ..... Air Force Research Laboratory            | ● GOOGLE ..... Google Headquarters                                | ● L&C ..... Lewis & Clark                           | ● NWU ..... North Western University                    |
| ● ANU ..... Australian National University            | ● GT ..... Georgia Tech   | ● LEEDS ..... University of Leeds                   | ● OXFORD ..... Oxford University                        |
| ● ARL ..... Army Research Lab                         | ● GWU ..... George Washington University                          | ● LLNL ..... Lawrence Livermore National Laboratory | ● PC ..... Pennsylvania State University                |
| ● AT&T ..... AT&T                                     | ● HP ..... Hewlett Packard  | ● LU ..... Lancaster University                     | ● PENN ..... Pennsylvania State University              |
| ● AUBURN ..... Auburn University                      | ● HPHC ..... Harvard Pilgrim Health Care Institute                | ● MIT ..... Massachusetts Institute of Technology   | ● PITT ..... Pennsylvania State University              |
| ● BC ..... Brooklyn College                           | ● ICSI ..... International Computer Science Institute at Berkeley | ● MITLL ..... MIT Lincoln Laboratory                | ● POTSDAM ..... State University of New York at Potsdam |
| ● BUAA ..... Beihang University                       | ● IIT ..... Illinois Institute of Technology                      | ● NAVY ..... United States Naval Academy            | ● PSU ..... Pennsylvania State University               |
| ● CCT ..... Clarkson University                       | ● IUT ..... Isfahan University of Technology                      | ● NCSU ..... North Carolina State University        | ● PU ..... Pennsylvania State University                |
| ● CMU ..... Carnegie Mellon University                | ● INL ..... Idaho National Laboratory                             | ● NDSU ..... North Dakota State University          | ● RICE ..... Rice University                            |
| ● CU ..... Cornell University                         | ● IU ..... Indiana University                                     | ● Newcastle(UK) ..... Newcastle University          | ● RIT ..... Rensselaer Institute of Technology          |
| ● DC ..... Dartmouth College                          | ● JWGU ..... Goethe University                                    | ● NLM-NIH ..... National Library of Medicine        | ● RPI ..... Rensselaer Institute of Technology          |
| ● EPFL ..... École Polytechnique Fédérale de Lausanne | ● KENT ..... University of Kent                                   | ● NOVA ..... Universidade Nova de Lisboa            | ● RSA ..... RSA Security                                |
| ● FUJITSU ..... Fujitsu North America Headquarters    | ● KTH ..... KTH Royal Institute of Technology                     | ● NSA ..... National Security Agency                | ● RU ..... Rutgers University                           |
| ● GMU ..... George Mason University                   |   | ● NU ..... Northeastern University                  | ● SC ..... South Carolina                               |



The Science of Security (SoS) Initiative at the NSA Research Directorate promotes foundational cybersecurity science that is needed to mature the cybersecurity discipline and to underpin advances in cyber defense. Beginning in 2012, one part of the initiative was to fund foundational research at labelets. With emphasis on building a community, each labelet created partnerships, or sub-labelets, with other universities. SoS researchers often freely collaborate with researchers in other institutions worldwide. In 2014, the SURE project was founded to investigate cybersecurity in the cyber-physical systems realm. This map illustrates the expansion of the SoS Initiative community.

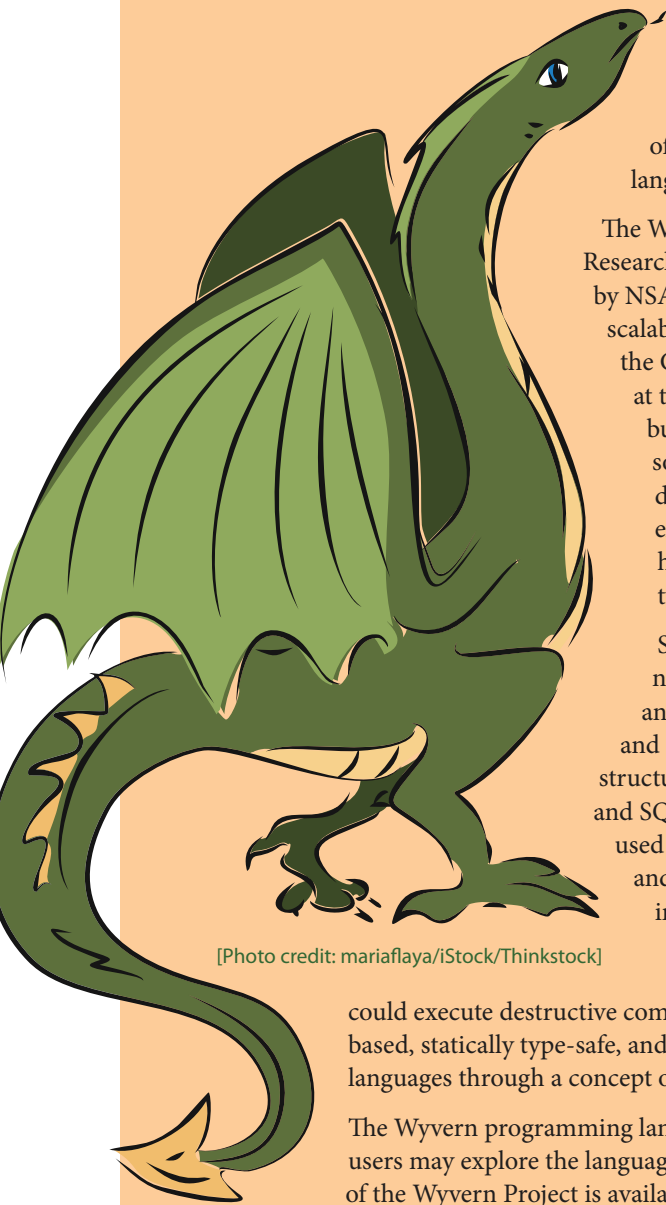


..... Northwestern University	● SIEMANS.....Siemens	● UIUC.....University of Illinois at Urbana-Champaign	● UTA.....University of Texas at Austin
..... University of Oxford	● SMU.....Singapore Management University	● ULISBOA.....University of Lisbon	● UTSA.....University of Texas at San Antonio
..... Providence College	● SU.....Stanford University	● UMD.....University of Maryland	● UVA.....University of Virginia
..... University of Pennsylvania	● Symantec.....Symantec Corporation	● UMich.....University of Michigan	● UVC.....University of Victoria
..... University of Pittsburgh	● THU.....Tsinghua University	● UNC.....University of North Carolina at Chapel Hill	● UW.....University of Washington
..... Potsdam University	● TX A&M.....Texas A&M	● UNCC.....University of North Carolina at Charlotte	● UW-Madison.....University of Wisconsin
..... Pennsylvania State University	● UA.....University of Alabama	● UNIMELB.....University of Melbourne	● UWAR.....University of Warwick
..... Purdue University	● UAlbany.....University of Albany	● UNL.....University of Nebraska Lincoln	● Verisign.....Verisign Labs
..... Rice University	● UC.....University of California at Oakland	● UNM.....University of New Mexico	● VT.....Virginia Polytechnic Institute
..... Rochester Institute of Technology	● UC Berkeley.....University of California at Berkeley	● UOFW.....University of Waterloo	● VU.....Vanderbilt University
..... Rensselaer Polytechnic Institute	● UCF.....University of Central Florida	● UP.....Universidade de Porto	● VUW.....Victoria University Wellington
..... RSA Security	● UDS.....Universitat des Saarlandes	● UPV.....Universitat Politècnica de Valencia	● WSU.....Wayne State University
..... Rutgers University	● UH Manoa.....University of Hawaii at Manoa	● USC.....University of Southern California	
..... Swarthmore College	● UIO.....University of Oslo	● USI.....Università della Svizzera Italiana (Switzerland)	

# POINTERS



## Wyvern programming language builds secure apps



[Photo credit: mariaflaya/iStock/Thinkstock]

In legend, the dragon-like creature known as the wyvern used its fiery breath and poisonous bite to protect its treasure from thieves. Still a popular figure on coats of arms and in electronic games, the wyvern has also inspired researchers at the Carnegie Mellon University (CMU) Science of Security (SoS) Lablet to develop the Wyvern extensible programming language to keep web and mobile applications more secure.

The Wyvern project, led by Dr. Jonathan Aldrich of the Institute for Software Research in the CMU School of Computer Science, is part of research supported by NSA and other agencies to address hard problems in cybersecurity, including scalability and composability. Dr. Aldrich, in collaboration with researchers at the CMU SoS Lablet, New Zealand colleague Dr. Alex Potanin, and researchers at the Victoria University of Wellington, have been developing Wyvern to build secure web and mobile applications. The language is designed to help software engineers build those secure applications using several type-based, domain-specific languages within the same program. Wyvern is able to exploit knowledge of sublanguages (e.g., structured query language (SQL), hypertext markup language (HTML), etc.) used in the program based on types and their context, which indicate the format and typing of the data.

Software development has come a long way, but the web and mobile arenas nonetheless struggle to cobble together different languages, file formats, and technologies. This proliferation is inefficient and thwarts scalability and composability. For example, a typical web page might require HTML for structure, cascading style sheets for design, JavaScript to handle user interaction, and SQL to access the database back-end. The diversity of languages and tools used to create an application increases the associated development time, cost, and security risks. It also creates openings for cross-site scripting and SQL injection attacks. Wyvern eliminates the need to use character strings as commands, as is the case, for instance, with SQL. By allowing character strings, malicious users with a rough knowledge of a system's structure

could execute destructive commands. Instead, Wyvern is a pure object-oriented language that is value-based, statically type-safe, and supports functional programming. It supports HTML, SQL, and other web languages through a concept of composable type-specific languages.

The Wyvern programming language is hosted at the open-source site GitHub; interested potential users may explore the language at <https://github.com/wyvernlang/wyvern>. In addition, a description of the Wyvern Project is available on the Cyber Physical Systems Virtual Organization web page at <http://cps-vo.org/node/21424>.

# Building secure and resilient software from the start

Researchers at the North Carolina State University (NCSU) SoS Labet are addressing the hard problems of resilience and predictive metrics in the development of secure software. The team, under principal investigators Laurie Williams and Mladen Vouk, empirically determined that security vulnerabilities (i.e., faults that violate implicit or explicit security policies) can be introduced into a system because the code is too complex, changed too often, or not changed appropriately. According to one NCSU study, source code files changed by nine developers or more were 16 times more likely to have at least one vulnerability after release. From such analyses, researchers can develop and disseminate predictive models and useful statistical associations to guide the development of secure software.

NCSU researchers have identified two key questions whose answers will require rigorous analysis and testing:

1. If highly attack-resilient components and appropriate attack sensors are developed, will it become possible to compose a resilient system from these component parts?

2. If so, how does that system scale and age?

One very simple and effective defensive strategy would be to build in a dynamic application firewall that does not respond to “odd” or out-of-norm inputs, such as those associated with zero-day attacks. While not foolproof, this approach would force attackers to operate within an application’s “normal” operational profile.

The research has generated tangible results. Three recent papers have been written as a result of this research. “On coverage-based attack profiles,” by Anthony Rivers, Mladen Vouk, and Laurie Williams (doi: 10.1109/SERE-C.2014.15); “A survey of common security vulnerabilities and corresponding countermeasures for SaaS,” by Donhoon Kim, Vouk, and Williams (doi: 10.1109/GLOCOMW.2014.7063386); and “Diversity-based detection of security anomalies” by Roopak Venkatakrishnan and Vouk (doi: 10.1145/2600176.2600205). For the complete SoS newsletter article, visit <http://www.cps-vo.org/node/21426>.

## Interest in cybersecurity science heats up at HotSoS 2015

The 2015 Symposium and Bootcamp on the Science of Security (HotSoS) took place April 21–22 at the UIUC National Center for Supercomputing Applications. This third annual conference, part of the NSA SoS project, brought together researchers from numerous disciplines seeking a rigorous scientific approach toward cyber threats. David Nicol, director of the Information Trust Institute and coprincipal investigator for the UIUC SoS Labet, was conference chair. Kathy Bogner, Intelligence Community coordinator for cybersecurity research, represented the NSA sponsor. Featured speakers included Mike Reiter, Lawrence M. Slifkin distinguished professor of computer science, University of North Carolina; Jonathan Spring,

researcher and analyst for the Computer Emergency Response Team division of the Software Engineering Institute, CMU; and Patrick McDaniel, professor of computer science and director of the Systems and Internet Infrastructure Security Laboratory, Penn State University.

Five tutorials and a workshop took place concurrently with individual presentations. Tutorials covered



[Photo credit: Sifis Diamantidis/Hemera/Thinkstock]



social network analysis; human behavior; policy-governed secure collaboration, security-metrics-driven evaluation, design, development and deployment; and resilient architectures. Thirteen individual papers, presented by British and US researchers, covered the following: signal intelligence analyst tasks, detecting abnormal user behavior, tracing cyber-attack analysis processes, vulnerability prediction models, preemptive intrusion detection, enabling forensics, global malware encounters, workflow resiliency, sanctions, password policies, resource-bounded systems integrity assurance, active cyber defense, and science of

trust. The agenda and presentations are available from the Cyber-Physical Systems Virtual Organization (CPS-VO) website. Members' may access the site at <http://cps-vo.org/node/3485/browser>. Nonmembers may access the site at <http://cps-vo.org/group/SoS>.

Next year's event will take place in Pittsburgh and be hosted by the CMU SoS Lablet.

# Adoption of Cybersecurity Technology Workshop

The Special Cyber Operations Research and Engineering (SCORE) subcommittee sponsored the 2015 Adoption of Cybersecurity Technology (ACT) Workshop at the Sandia National Laboratories in Albuquerque, New Mexico, from 3–5 March 2015. The vision for the workshop was to change business practices for adoption of cybersecurity technologies; expose developers, decision makers, and implementers to others' perspectives; address the technology, process, and usability roadblocks to adoption; and build a community of interest to engage regularly.

Attendees represented four segments of the cybersecurity world: researchers and developers, decision makers, implementers, and experts on human behavior. They explored, developed, and implemented action plans for four use cases that addressed each of the four fundamental cybersecurity goals: 1) device integrity, 2) authentication and credential protection/defense of accounts, 3) damage containment, and 4) secure and available transport. This construct provided a framework specifically designed to confront spear phishing.

Participants were primarily government personnel, with some individuals from federally funded research and development centers, academia, and industry. These cybersecurity professionals believe that as many as 80% of their field's current problems have known solutions that have not been implemented.

The workshop is the kickoff activity for a sustained effort to implement cybersecurity technology solutions throughout the US government, and it is expected to become an annual event.

The agenda focused on specific threat scenarios, cohorts' concerns to promote understanding among groups, addressing the use cases, and developing action plans for implementation via 90-day phases known as "spins." Participants will brief the spin results to the ACT organizing committee and to threat analysts who will assess progress.

In order to illuminate systemic barriers to adoption of security measures, the workshop focused specifically on countering the threat from spear phishing, a social-engineering trick that hackers use to convince people to provide passwords, financial data, and other private information.

The goal over the next year is to strengthen government networks against spear phishing attacks by applying the selected technologies identified in the four use cases. Although this activity is tactical in nature, it provides an underlying strategy for achieving broader objectives as well as a foundation for transitioning collaborative cybersecurity engagements.

For the complete article, visit: <http://www.cps-vo.org/node/21405>.

# Sandboxing in the cloud

Cybersecurity experts working with complex systems often have to isolate certain components that cannot be fully trusted. A common technique for this is known as sandboxing—a method in which security layers are put in place to encapsulate software components and impose policies that regulate interactions between the isolated components and the rest of the system. Sandboxes provide a mechanism to work with a full sampling of components and applications, even when they are known to be malicious. Of course, if the sandbox fails or is bypassed, the production environment may be compromised.

Michael Maass, William L. Scherlis, and Jonathan Aldrich from the Carnegie Mellon University (CMU) Institute for Software Research in the School of Computer Science propose a cloud-based sandbox method to mitigate the risk of sandbox failure and raise the bar for attackers. The researchers call their approach “in-nimbo sandboxing,” after *nimbus*, the Latin word for “cloud.” Using a technique they liken to software-as-a-service, they tailor the sandbox to the specific application in order to encapsulate components with smaller, more defensible attack surfaces than other techniques. This remote encapsulation reduces the likelihood of a successful attack as well as the magnitude or degree of damage from a successful attack.

The authors note that “most mainstream sandboxing techniques are in-situ, meaning they impose security policies using only trusted computing bases within the system being defended. Existing in-situ sandboxing approaches decrease the risk that a vulnerability will be successfully exploited because they force the attacker to chain multiple vulnerabilities together or bypass the sandbox. Unfortunately, in practice, these techniques still leave a significant attack surface, leading to a number of attacks that succeed in defeating the sandbox.”

With funding from the CMU Science of Security (SoS) Lablet, Maass, Scherlis, and Aldrich conducted a field trial with a major aerospace firm. They were able to compare an encapsulated component deployed in an enterprise-managed cloud, with the original, unencapsulated component deployed in the relatively higher-value user environment. The researchers’ assessment was based on the criteria of performance, usability, and security.

1. **Performance evaluation:** The trial focused on latency of interactions and ignored resource consumption. For deployed applications, the technique increased user-perceived latency only slightly.
2. **Usability evaluation:** The sandbox’s design was structured to present an experience identical to the local version, and users judged that this was accomplished. Results suggest that the in-nimbo approach may be feasible for other types of systems as well because a field trial system is built primarily using widely adopted established components.
3. **Security evaluation:** Cloud-based sandboxes are more difficult to attack, partly because defenders can customize the environment in which an encapsulated computation takes place and partly because of the inherently ephemeral nature of cloud-computing environments. The in-nimbo system separates a component of interest from its operating environment and replaces it with a specialized transduction mechanism to manage interactions with the now-remote component, which has been moved to the cloud environment.

The authors indicate that this work is a precursor to an extensive study, still in progress, that evaluates more than 70 examples of sandbox designs and implementations against a range of identified criteria. The full study, “In-nimbo sandboxing,” can be accessed at <http://doi.acm.org/10.1145/2600176.2600177>. For the SoS newsletter article, visit <http://www.cps-vo.org/node/21422>.



[Photo credits: Lordn, mipan/iStock/Thinkstock]

# Science of SecUrity and REsilience for Cyber-Physical Systems project

On 17–18 March 2015, the NSA Trusted Systems Research Group met in Nashville, Tennessee with academic researchers from the Science of SecUrity and REsilience for Cyber-Physical Systems (SURE) project to review their first six months of work. Researchers came from four participating institutions—Vanderbilt University, the University of Hawai'i, the University of California at Berkeley, and the Massachusetts Institute of Technology (MIT). Individuals from the National Science Foundation, Nuclear Regulatory Commission, and Air Force Research Labs also attended.

SURE is the NSA-funded project aimed at improving scientific understanding of “resiliency”; that is, the robustness of a cyber-physical system (CPS) to reliability failures or faults, as well as survivability against security failures and attacks. Initially, SURE focused on CPS architectures related to only water distribution and surface traffic control; the project now also focuses on air traffic control and satellite systems. The principal investigator for the SURE project is Xenofon Koutsoukos, professor of electrical engineering, computer science, and senior research scientist in the Institute for Software Integrated Systems at Vanderbilt University. Professor Koutsoukos indicated the use of these additional CPSs is to demonstrate how the SURE methodologies can apply to multiple systems.

The SURE project addresses the question of how to design systems that are resilient despite significant decentralization of resources and decision-making. Main research thrusts include hierarchical coordination and control, science of decentralized security, reliable and practical reasoning about secure computation and communication, evaluation and experimentation, and education and outreach.

The Resilient Cyber Physical Systems (RCPS) test bed, discussed in Emfinger, Kumar, and Karsai's article in this issue (“Resilient and secure cyber-physical systems”), supports evaluation and experimentation across the entire SURE research portfolio.

In addition to the RCPS test bed, researchers presented 10 resiliency projects on behavioral and technical subjects including adversarial risk, active learning for malware detection, privacy modeling, actor networks, flow networks, control systems, software and software architecture, and information flow policies. The scope and format of the Cyber Physical Systems Virtual Organization web site (<http://cps-vo.org>) was also briefed. For the complete Science of Security newsletter article, visit <http://www.cps-vo.org/node/21425>.



[Photo credit: Nerthuz/iStock/Thinkstock]



# Selecting Android graphic pattern passwords

With funding from the University of Maryland Science of Security Lablet, researchers at the US Naval Academy and Swarthmore College conducted a large study of user preferences relating to usability and security of graphical password patterns used to access Android mobile phones.

Android mobile phones come with four embedded access methods: a finger swipe, a pattern, a PIN, or a password, in ascending order of security. In the pattern method, the user is required to select a pattern by traversing a grid of 3 x 3 points. A pattern must contain at least four points, cannot use a point twice, and all points along a path must be sequential and connected; that is, no points along the path may be skipped. The pattern can be visible or cloaked.

When a user enables such a feature, however, how does that person trade off security with usability? Also, are there visual cues that lead users to select one password over another and whether for usability or security?

The study by Adam Aviv and Dane Fichter, "Understanding visual perceptions of usability and security of Android's graphical password pattern," uses a survey methodology that asks participants to select between pairs of patterns and indicate either a security or usability preference. By carefully selecting password pairs to isolate a visual feature, a visual perception of usability and security of different features were measured. The 384 users in the study sample were self-selected via Amazon Mechanical Turk, an online marketplace for crowdsourcing. Users found that visual features that can be attributed to complexity indicated a stronger perception of security, while spatial features, such as shifts up/down or left/right are not strong indicators either for security or usability.

In their study, Aviv and Fichter selected pairs of patterns based on six visual features:

1. **Length:** Total number of contact points used
2. **Crosses:** The pattern double-backs on itself by tracing over a previously contacted point
3. **Nonadjacent:** The total number of nonadjacent swipes which occur when the pattern double-backs on itself by tracing over a previously contacted point

4. **Knight-moves:** Moving two spaces in one direction and then one space over in another direction, like the knight piece in chess
5. **Height:** The amount the pattern is shifted towards the upper or lower contact points
6. **Side:** The amount the pattern is shifted towards the left or right contact points

Users were asked to choose between two passwords, indicating a preference for one password meeting a particular criterion (such as perceived security or usability) over the other password. By carefully selecting these password pairs, researchers could isolate the passwords' visual features and measure the impact of a particular feature on users' perception of security and usability.

The researchers concluded that spatial features have little impact. More visually striking features have a stronger impact, with the length of the pattern being the strongest indicator of preference. These results were extended and applied by constructing a predictive model with a broader set of features from related work, and they revealed that the distance factor, the total length of all the lines in a pattern, is the strongest predictor of preference. These findings provide insight into users' visual calculus when assessing a password, and this information may be used to develop new password systems or user selection tools, like password meters.

Moreover, Aviv and Fichter conclude that, with a good



[Photo credit: Kirillm/iStock/Thinkstock]

predictive model of user preference, this research could be expanded and these findings could be applied to a broader set of passwords. For example, ranking data based on learned pairwise preferences is an active research area in machine learning, and the resulting rankings metric over all potential patterns in the space would greatly benefit

the security community. For example, such a metric could enable new password selection procedures. The full study is available at <http://www.usna.edu/Users/cs/aviv/papers/p286-aviv.pdf>. For the complete Science of Security newsletter article, visit <http://www.cps-vo.org/node/21423>.

# Computational Cybersecurity in Compromised Environments Workshop

The Special Cyber Operations Research and Engineering (SCORE) subcommittee sponsored the 2014 Computational Cybersecurity in Compromised Environments (C3E) Workshop at the Georgia Tech Research Institute (GTRI) Conference Center in Atlanta from 20–22 October 2014. This event, the sixth in a series of annual C3E workshops, brought together top academic, industry, and government experts to examine new ways of approaching the nation's cybersecurity challenges. In particular, participants discussed how to enable smart, real-time decision-making in situations involving both “normal” complexity and persistent adversarial behavior.

Since its start in 2009, C3E's overarching objectives have been to develop ideas worth additional research and to develop a community of interest around unique, analytic, and operational approaches to the persistent threat.

C3E 2014 continued to focus on the needs of the practitioner and leverage past themes of predictive analytics, decision-making, consequences, and visualization. The two tracks, Security by Default and Data Integrity, were developed based on recommendations from senior government officials in order to ensure that outcomes will be applicable to real-world security needs. As in previous years, the event included keynote addresses on a wide range of cybersecurity topics, a discovery or challenge problem that attracted participants' analytic attention prior to the workshop, and track work that involved small-group focus on security by default and data integrity.

This year's discovery problem focused on approaches that challenge traditional thinking on using metadata to identify high-interest, suspicious, and likely malicious behaviors. Participants used the Department of Homeland Security's Protected Repository for the Defense of

Infrastructure against Cyber Threats (PREDICT), a rich collection including the borderline gateway protocol, the domain name system, data applications (net analyzer), infrastructure (census probe), and security (botnet sinkhole data, dark data, etc.).

The Data Integrity Track addressed issues associated with finance and health/science, and captured relevant characteristics. Track participants also identified potential solutions and research themes for addressing data integrity issues: 1) diverse workflows and sensor paths, 2) cyber insurance and regulations, and 3) human-in-the-loop data integrity detection.

The Security by Default Track focused on addressing whether it is possible to create systems that are secure when they are fielded, and examined the common perception among stakeholders that such systems may be less functional, less flexible, and more difficult to use. Participants identified five topics that might merit additional study:

1. The “building code” analogy—balancing the equities,
2. Architecture and design—expressing security problems understandably,
3. Architecture and design—delivering and assuring secure components and infrastructure,
4. Biological analogs—supporting adaptiveness and architectural dynamism, and
5. Usability and metrics—rethinking the trade-off between security and usability.

For the complete article, visit <http://www.cps-vo.org/node/21421>.

# FROM LAB TO MARKET

News from the Technology Transfer Program

## NSA shares cyber software via open source

In June, the National Security Agency uploaded the Security Integrity Management Platform (SIMP) software to its new open-source software (OSS) site on GitHub, one of the world's largest hosting services for sharing source code. This technology can provide a solid "defense-in-depth" strategy for enterprise security and will make it easier for other government organizations and private industry to adopt this tool to help fortify their networks against cyber threats.

SIMP enables the creation of secured Linux clusters that can maintain a specific security posture and ensure system integrity by consistently applying policy-oriented and best-practice configurations over time. SIMP incorporates the centralized management tool from Puppet Labs to address process degradation at the operating-system level and eases auditing, accreditation, and client upgrade processes. SIMP is a continually managed, minimal Linux distribution based on Red Hat 6.x and 7.x; it can also support the Community Enterprise Operating System (CentOS) Linux distribution.

NSA has increasingly shared its research with the open-source software community, a move that can accelerate the development of innovative capabilities and solutions in both the public and private sectors. For example, the NSA Technology Transfer Program (TTP) recently prepared the OSS release of the Agency's NiagaraFiles (Nifi) technology via the Apache Software Foundation. The global reviews and critiques that stem from open-source releases can broaden a technology's applications for the US private sector and for the good of the nation at large.

Why does NSA need its own GitHub site? Linda Burger, director of the NSA TTP, notes that "an agency site aggregates the niche, cutting-edge

technology that NSA is offering to open source via GitHub, making it easy to find and access."

Burger also noted OSS releases are extremely efficient for transferring technology from the federal laboratory to the marketplace. "This approach enables the Agency to adapt to the tremendous speed of technology advancement in support of mission solutions, while also making applications available to the public at no charge."

Carl Caum, senior technical marketing manager at Puppet Labs, said his company is excited by NSA's contribution to the open source community.

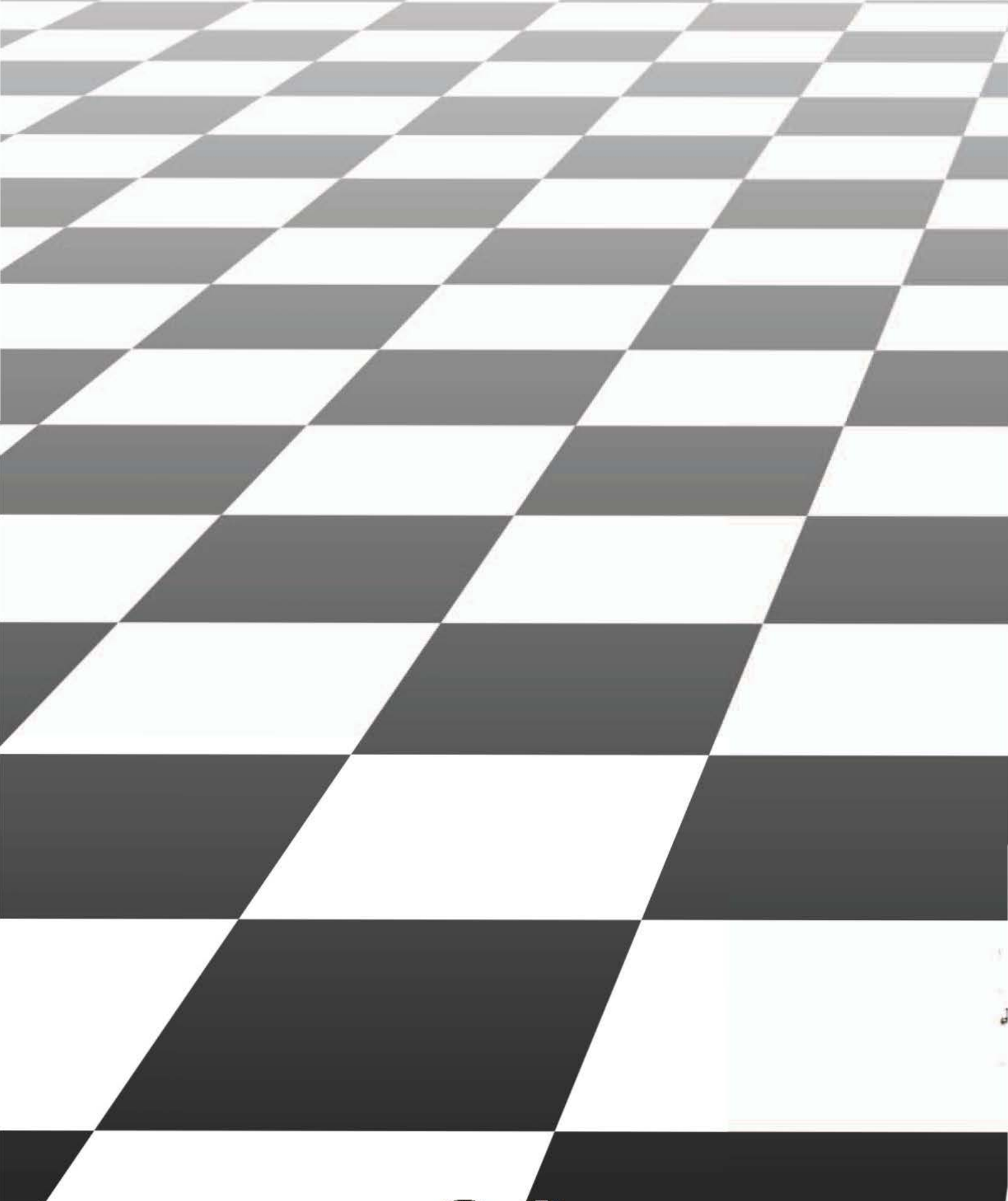
"The goal of the SIMP framework is to help users automatically enforce compliance with various Security Content Automation Program (SCAP) profiles through the consistent configuration of core infrastructure components. NSA's contributions can help other organizations with similar challenges, so operations teams don't have to spend precious time trying to solve the same problem themselves. That's the power and beauty of open-source software."

To learn more about technology transfer activities at NSA, please contact the TTP at [tech\\_transfer@nsa.gov](mailto:tech_transfer@nsa.gov) or 1-866-680-4539. 



[Photo credit: cacaroot/iStock/Thinkstock]





NATIONAL SECURITY AGENCY



CENTRAL SECURITY SERVICE

*Defending Our Nation. Securing The Future*