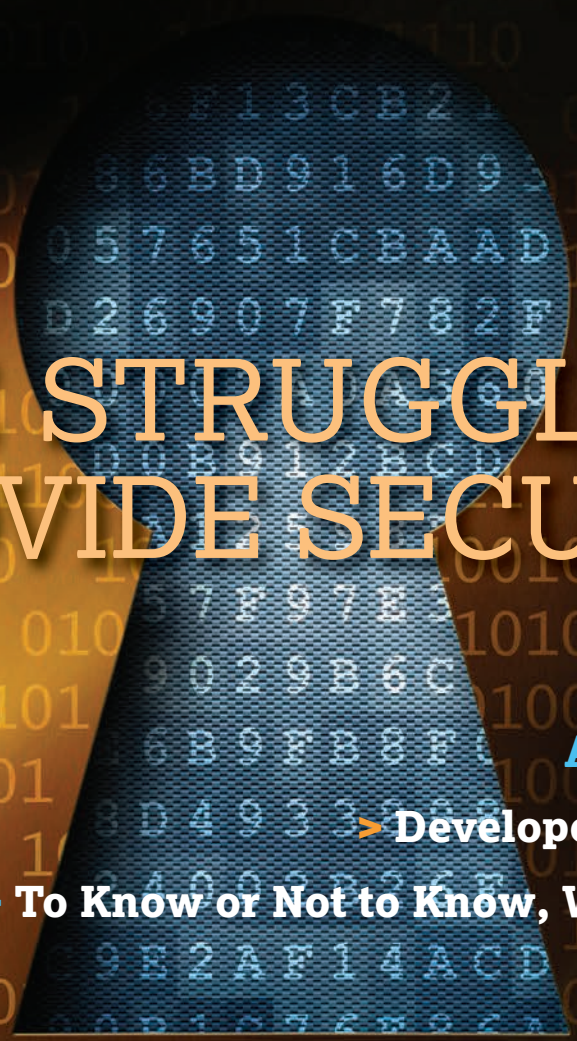
COMPUTING edge



THE STRUGGLE TO PROVIDE SECURITY

Also in this issue:

> Developer, Debug Thyself

> To Know or Not to Know, What is the Need?

MARCH 2016

www.computer.org

IEEE computer society

CELEBRATING 70 YEARS

IEEE computer society

ROCK STARS OF RISK-BASED SECURITY

Learn What You Must Know About Risk Assessment and Mitigation

12 April 2016 | Washington, DC Metro Area

100% Security Solution? Pipedream!

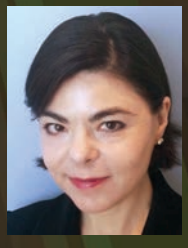
Virtually every company will be hacked, and today, experts accept that a 100% security solution is not feasible. Advanced risk assessment and mitigation is the order of the day.

Rock Stars of Risk-Based Security is the must attend symposium of its kind in 2016 on this critical new reality. What attacks can you expect? How can you be prepared? On April 12, 2016 you'll learn the answers to those questions straight from the people who are driving innovation in risk-based security.

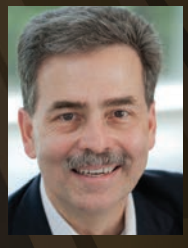
#RSRBSeast

Rock Star Speakers

Scott Borg
Director (CEO) and Chief Economist, U.S. Cyber Consequences Unit



Diana Kelly
Executive Security Advisor, IBM



Sam Phillips
Vice President and General Manager of Security Services and Chief Information Security Officer, Samsung

www.computer.org/rbseast

COMPUTING edge

IEEE COMPUTER SOCIETY http://computer.org • +1 714 821 8380

Printed with inks containing soy and/or vegetable oils

IEEE computer society
CELEBRATING 70 YEARS

STAFF

BPA WORLDWIDE

Circulation: ComputingEdge (ISSN 2469-7087) is published monthly by the IEEE Computer Society. IEEE Headquarters, Three Park Avenue, 17th Floor, New York, NY 10016-5997; IEEE Computer Society Publications Office, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720; voice +1 714 821 8380; fax +1 714 821 4010; IEEE Computer Society Headquarters, 2001 L Street NW, Suite 700, Washington, DC 20036.

Postmaster: Send address changes to ComputingEdge-IEEE Membership Processing Dept., 445 Hoes Lane, Piscataway, NJ 08855. Periodicals Postage Paid at New York, New York, and at additional mailing offices. Printed in USA.

IEEE Computer Society Magazine Editors in Chief

Computer
Sumi Helal, *University of Florida*

IEEE Software
Diomidis Spinellis, *Athens University of Economics and Business*

IEEE Internet Computing
M. Brian Blake, *University of Miami*

IT Professional
San Murugesan, *BRITE Professional Services*

IEEE Security & Privacy
Ahmad-Reza Sadeghi, *Technical University of Darmstadt*

IEEE Micro
Lieven Eeckhout, *Ghent University*

IEEE Computer Graphics and Applications
L. Miguel Encarnação, *ACT, Inc.*

IEEE Pervasive Computing
Maria Ebling, *IBM T.J. Watson Research Center*

Computing in Science & Engineering
George K. Thiruvathukal, *Loyola University Chicago*

IEEE Intelligent Systems
Daniel Zeng, *University of Arizona*

IEEE MultiMedia
Yong Rui, *Microsoft Research*

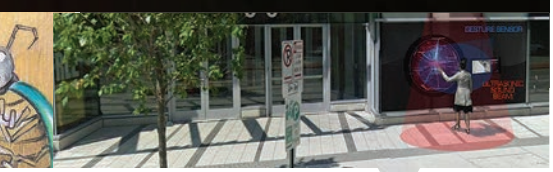
IEEE Annals of the History of Computing
Nathan Ensmenger, *Indiana University Bloomington*

IEEE Cloud Computing
Mazin Yousif, *T-Systems International*

MARCH 2016 • VOLUME 2, NUMBER 3

COMPUTING edge

Departments



42

Small Data, Big Impact

Securing Mobile Applications

Elisa Bertino, Purdue University

This installment highlighting the work published in IEEE Computer Society journals comes from IEEE Transactions on Dependable and Secure Computing.

Widespread mobile device use has stimulated a rich market for applications. Many apps, however, reveal sensitive user information such as location, movements, and habits[1] and/or spread malware.[2]

Network anonymization techniques alone don't ensure privacy because the OS together with the invoked mobile apps might still release information that reidentifies users or devices. Even when users are careful not to provide identifying data to smartphone apps over anonymous connections, the apps can leak such information without user knowledge. Thus, we must devise accurate methods of checking apps for the presence of malware and spyware.

Although third-party application markets exist, most users download apps from well-known markets such as Google Play, Amazon Appstore, iTunes App Store, and Windows Store. The availability and widespread use of these markets might allow centralized deployment of techniques that identify potentially malicious apps.

Uncovering potentially malicious apps isn't a trivial task. Proposed approaches typically differ in the features they use to identify such apps, their use of machine-learning techniques, and their accuracy.

In their 2015 *IEEE Transactions on Dependable and Secure Computing* article, Lei Cen and his colleagues proposed a highly accurate model for detecting malware in Android apps.[3] The authors observed that application markets distribute apps in a form that allows easy decompilation and thus analysis. Moreover, they noted that mobile platforms provide semantically rich APIs. Drawing on these two observations, the authors devised a discriminative probabilistic learning model, based on regularized logistic regression, that detects malware by using apps' decompiled code and information about required permissions.

Despite this important breakthrough, more work is needed on mobile application security. Cen and his colleagues' technique is static and thus doesn't protect apps compromised after being uploaded to a mobile device. Syed Hussain and his colleagues recently proposed an approach that monitors database apps for anomalous behaviors.[4] A promising research direction would be to apply a similar method to mobile device apps. By using static analysis, profiles of apps' expected behavior could be created and then monitored at runtime for anomalies.

REFERENCES

1. B. Shebaro et al., "IdentiDroid: Android Can Finally Wear Its Anonymous Suit," *Trans. Data Privacy*, vol. 7, no. 1, 2014, pp. 27–50.
2. S.-H. Seo et al., "Detecting Mobile Malware Threats to Homeland Security through Static Analysis," *J. Network and Computer Applications*, vol. 38, 2014, pp. 43–53.
3. L. Cen et al., "A Probabilistic Discriminative Model for Android Malware Detection with Decompiled Source Code," *IEEE Trans. Dependable and Secure Computing*, vol. 12, no. 4, 2015, pp. 400–412.
4. S.R. Hussain, A. Sallam, and E. Bertino, "DetAnom: Detecting Anomalous Database Transactions by Insiders," *Proc. 5th ACM Conf. Data and Application Security and Privacy* (CODASPY 15), 2015, pp. 25–35.

ELISA BERTINO is a professor of computer science, the director of the Cyber Center, and the research director of the Center for Education and Research in Information Assurance and Security at Purdue University. Contact her at bertino@cs.purdue.edu.

Selected CS articles and columns are also available for free at **http://ComputingNow.computer.org**.

 Published by the IEEE Computer Society

Magazine Roundup

The IEEE Computer Society's lineup of 13 peer-reviewed technical magazines covers cutting-edge topics ranging from software design and computer graphics to Internet computing and security, from scientific applications and machine intelligence to cloud migration and microchip manufacturing. Here are highlights from recent issues.

Computer

Computer's February 2016 special issue explores compelling new **cloud-computing trends and advances**. Cloud computing research—once focused on the management of virtual machines and provision of infrastructure as a service—now includes virtually all aspects of modern distributed computing. Article topics in this issue include an approach that software developers can use to weigh the pros and cons of moving their applications to the cloud, ways to manage trust in the cloud, and the evaluation of cloud resource-orchestration frameworks.

IEEE Software

Misbehaving software has been around for decades but is now so widespread that developers must take this problem more seriously and find solutions, according to "Developer, Debug Thyself," from *IEEE Software*'s January/February 2016 issue.

IEEE Internet Computing

The Internet has facilitated previously unimaginable economic applications, including targeted online advertising, crowdsourcing, peer-to-peer lending, and digital currencies. *IEEE Internet Computing*'s January/February 2016 special issue on **Internet economics** discusses the modeling, analysis, and design of Internet-specific economic activity and the research challenges this entails.

Computing in Science & Engineering

Database management systems have become an indispensable tool for industry, government, and academia, and form a significant component of modern datacenters. "**Hardware Acceleration for Query Processing**: Leveraging FPGAs, CPUs, and Memory," "which appears in *CiSE*'s January/February 2016 issue, looks at the organization of a modern relational database management system and proposes optimizations and redesigns in several areas.

IEEE Security & Privacy

The safety and dependability of software is just as important as

2376-113X/16/$33.00 © 2016 IEEE Published by the IEEE Computer Society

its security and privacy. *IEEE S&P*'s January/February 2016 special issue on **software everywhere** focuses on these four aspects of the technology, examining issues such as ethics, open source software, and the Internet of Things.

IEEE Cloud Computing

The social Internet of Things is creating unprecedented online and offline social experiences and is changing our social patterns. The social cloud has the potential to use social relationships to improve resource sharing in social networks. Both fields require more research, which is examined by *IEEE Cloud Computing*'s November/December 2015 special issue on **enabling the social Internet of Things and social cloud**.

IEEE Computer Graphics and Applications

CG&A's January/February 2016 special issue showcases articles on **using computing environments to improve the human experience**. A common element is adding touch to visual representations, long established as necessary for improving human–computer interaction. The articles look at different approaches to improving, mimicking, or facilitating the need for human touch in digital experiences.

IEEE Intelligent Systems

Online behavioral analysis and modeling has aroused considerable interest from professionals in closely related research fields such as data mining, machine learning, and information retrieval. *IEEE Intelligent Systems*' January/February 2016 special issue provides a forum for behavioral-analysis researchers to review pressing needs, discuss challenging research issues, and showcase state-of-the-art R&D in modern Web platforms.

IEEE MultiMedia

In recent years, the number of social-media services—which facilitate a more socially connected Web—has increased. At the same time, the growth of massively open online courses (MOOCs) has made previously expensive educational resources available to learners throughout the world. Using these resources effectively via social-media services is the topic of *IEEE MultiMedia*'s January–March 2016 special issue on **social media for learning**.

IEEE Annals of the History of Computing

The beginning of Internet development in Latin America's largest country is the topic of "**The Dawn of the Internet in Brazil**," from *IEEE Annals*' October–December 2015 issue.

IEEE Pervasive Computing

During the past few decades, cars have transformed into complex computational platforms, underpinned by key intelligent-systems elements. *IEEE Pervasive Computing*'s January–March 2016 special issue offers insight into what we can expect from **smart vehicle spaces** and the technology that enables them.

IT Professional

In power grids, demand–response mechanisms integrate demand management into the power-scheduling process. IT will play a crucial role in managing such mechanisms for datacenters, allowing more efficient scheduling, according to "IT-Driven Power Grid Demand Response for Datacenters," from *IT Pro*'s January/February 2016 special issue on **cyberspace and energy management**.

IEEE Micro

The authors of "Architectural Simulators Considered Harmful," which appears in *IEEE Micro*'s November/December 2015 issue, examine the detrimental effect of overreliance on **quantitative hardware simulators**. They describe three broad pitfalls of simulators and simulator use, discuss how to avoid them, and propose a way to recalibrate evaluation standards.

Computing Now

The Computing Now website (http://computingnow.computer.org) features **up-to-the-minute computing news** and blogs, along with articles ranging from peer-reviewed research to opinion pieces by industry leaders.

Selected CS articles and columns are also available for free at http://ComputingNow.computer.org.

The Struggle to Provide Security

Organizations are gathering an ever-increasing amount of sensitive data about customers, medical patients, research projects, and other sources. As the amount of collected data grows, securing it becomes progressively harder.

Meanwhile, so many computing systems, smartphones, and even everyday objects are now connected to the Internet and to one another that hackers have more attack vectors than ever before.

These factors make it difficult to provide the security necessary to protect data, systems, individuals, and organizations. This *ComputingEdge* issue examines some of today's important cybersecurity issues and challenges.

Computer's "Four Software Security Findings" discusses key results from an analysis of 78 firms—based on the Building Security in Maturity Model—that could help firms protect and secure their assets.

"Biological Warfare: Tampering with Implantable Medical Devices," from *IT Professional*, explores the ability of hackers to attack implantable medical devices, such as cardiac pacemakers, and how to address this potentially deadly threat.

IEEE Security & Privacy's "Colluding Apps: Tomorrow's Mobile Malware Threat" looks at dealing with the problems that could occur if groups of mobile apps work together to play a small, undetectable role in a larger malicious operation.

"Managing Risk in a Cloud Ecosystem," from *IEEE Cloud Computing*, focuses on security risks related to the use of cloud-based information systems.

IT Professional's "Protecting Digital Assets: Legal Protections Do Not Equal Practical Security" examines how traditional legal protections don't meet contemporary digital-property owners' needs.

IEEE Internet Computing's "The Research Horizon: Four Nearly Practical Concepts" outlines four examples of what could be the next big influence on cybersecurity.

The best chance for getting security right is to make the cryptography as simple and public as possible, according to Bruce Schneier in *IEEE Security & Privacy*'s "Cryptography Is Harder than It Looks."

ComputingEdge articles on other subjects include the following:

- "Developer, Debug Thyself," from *IEEE Software*, says the risks of misbehaving software have become so widespread that the industry must take corrective action now.
- In a world of big data, sometimes having the right amount of small data at the right time can also pack a powerful punch, a concept explored in *IEEE MultiMedia*'s "Small Data, Big Impact."
- "To Know or Not to Know, What Is the Need?," from *IEEE Intelligent Systems*, discusses a new "need to know" principle for developing intelligent cyberdefense systems.

2376-113X/16/$33.00 © 2016 IEEE Published by the IEEE Computer Society

Four Software Security Findings

Gary McGraw, Cigital

Analyzing data from 78 firms using the Building Security In Maturity Model (BSIMM) revealed four truths about software security that will help firms protect and secure their assets.

Software security continues to grow and evolve, currently accounting for more than 10 percent of global IT security revenue worldwide. On the surface, it seems obvious that we must make software systems secure from the start, but opinions vary as to implementation. Through a multiyear process of observing and measuring security initiatives, we can move beyond opinion into the realm of fact. What follows are four indisputable facts we learned about software security through our work with the Building Security In Maturity Model (BSIMM; http://bsimm.com).

THERE'S NO SPECIAL SNOWFLAKE

The BSIMM is an observation-based study of software security that began in 2008 with nine firms. The sixth release (BSIMM6) includes data gathered from 78 firms, including Adobe, Aetna, Bank of America, Experian, Fannie Mae, Fidelity, Intel, LinkedIn, McAfee, PayPal, Siemens, Sony Mobile, Symantec, Visa, VMware, Wells Fargo, and Zephyr Healthcare. This latest data set is more than 20 times larger than it was in 2008.

Using the BSIMM measurement tool, a firm can directly compare its software security approach to the BSIMM community through 112 well-defined activities—for example, performing design review for high-risk applications—organized in 12 practices. One way to represent this measurement is shown in Figure 1, which illustrates how a target firm can be scored in a high-resolution fashion using the BSIMM scorecard.

DISCLAIMER

Some of the material in this article is used by permission of BSIMM coauthors Gary McGraw, Sammy Migues, and Jacob West. Download the BSIMM at http://bsimm.com.

 Published by the IEEE Computer Society 2376-113X/16/$33.00 © 2016 IEEE

EDITOR **JEFFREY VOAS**
National Institute of Standards and Technology; j.voas@ieee.org

The BSIMM model has been used to measure more than 110 firms to date, and many firms have been measured multiple times over several years. BSIMM measurements take the form of intensive in-person interviews with various stakeholders in a firm's software security initiative. A typical measurement process—including data gathering and analysis—takes two or three weeks to complete and results in a formal report.

The BSIMM community includes firms of various sizes in different industry verticals and with a range of levels of software security maturity. We've never come across a firm that couldn't be measured with the BSIMM—in other words, there's no special snowflake.

To give you some idea of the breadth of BSIMM6's coverage, consider that the model itself describes the work of 3,195 full-time software security professionals attempting to control the security of software developed by 287,006 developers building and evolving 69,750 applications. The BSIMM6 community includes 33 financial services firms, 27 independent software vendors, 13 consumer electronics firms, and 10 healthcare firms. Development groups in the BSIMM community range from a small group of 23 developers to a large population of 35,000 developers, with a median of 1,200 developers in a typical firm.

BSIMM SCORECARD FOR: FIRM | OBSERVATIONS: 37

GOVERNANCE			INTELLIGENCE			SSDL TOUCHPOINTS			DEPLOYMENT		
ACTIVITY	BSIMM6 FIRMS	FIRM	ACTIVITY	BSIMM6 FIRMS	FIRM	ACTIVITY	BSIMM6 FIRMS	FIRM	ACTIVITY	BSIMM6 FIRMS	FIRM
STRATEGY & METRICS			ATTACK MODELS			ARCHITECTURE ANALYSIS			PENETRATION TESTING		
[SM1.1]	41	1	[AM1.1]	17	1	[AA1.1]	67	1	[PT1.1]	69	1
[SM1.2]	40		[AM1.2]	51		[AA1.2]	29	1	[PT1.2]	47	1
[SM1.3]	36	1	[AM1.3]	31		[AA1.3]	22	1	[PT1.3]	47	
[SM1.4]	66	1	[AM1.4]	8	1	[AA1.4]	46		[PT2.2]	20	1
[SM2.1]	36		[AM1.5]	46	1	[AA2.1]	12		[PT2.3]	17	
[SM2.2]	29		[AM1.6]	11		[AA2.2]	9	1	[PT3.1]	10	1
[SM2.3]	30		[AM2.1]	6		[AA2.3]	13		[PT3.2]	8	
[SM2.5]	17		[AM2.2]	8	1	[AA3.1]	6				
[SM2.6]	29		[AM3.1]	4		[AA3.2]	1				
[SM3.1]	15		[AM3.2]	2							
[SM3.2]	7										
COMPLIANCE & POLICY			SECURITY FEATURES & DESIGN			CODE REVIEW			SOFTWARE ENVIRONMENT		
[CP1.1]	45	1	[SFD1.1]	61		[CR1.1]	18		[SE1.1]	37	
[CP1.2]	61		[SFD1.2]	59	1	[CR1.2]	53	1	[SE1.2]	69	1
[CP1.3]	41	1	[SFD2.1]	24		[CR1.4]	55	1	[SE2.2]	31	1
[CP2.1]	19		[SFD2.2]	39		[CR1.5]	24		[SE2.4]	25	
[CP2.2]	23		[SFD3.1]	8		[CR1.6]	27	1	[SE3.2]	10	
[CP2.3]	25		[SFD3.2]	11		[CR2.2]	7		[SE3.3]	5	
[CP2.4]	29		[SFD3.3]	2		[CR2.5]	20				
[CP2.5]	33	1				[CR2.6]	16				
[CP3.1]	18					[CR3.2]	3	1			
[CP3.2]	11					[CR3.3]	5				
[CP3.3]	6					[CR3.4]	3				
TRAINING			STANDARDS & REQUIREMENTS			SECURITY TESTING			CONFIG. MGMT & VULN. MGMT		
[T1.1]	59	1	[SR 1.1]	57	1	[ST1.1]	61	1	[CMVM1.1]	71	1
[T1.5]	26		[SR1.2]	50		[ST1.3]	66	1	[CMVM1.2]	73	
[T1.6]	17	1	[SR1.3]	52	1	[ST2.1]	24	1	[CMVM2.1]	64	1
[T1.7]	36		[SR2.2]	27		[ST2.4]	8		[CMVM2.2]	61	
[T2.5]	10		[SR2.3]	21		[ST2.5]	10		[CMVM2.3]	31	
[T2.6]	15	1	[SR2.4]	19		[ST2.6]	11		[CMVM3.1]	4	
[T2.7]	6		[SR2.5]	20	1	[ST3.3]	4		[CMVM3.2]	6	
[T3.1]	3		[SR2.6]	23	1	[ST3.4]	4		[CMVM3.3]	6	
[T3.2]	3		[SR3.1]	6		[ST3.5]	5		[CMVM3.4]	3	
[T3.3]	3		[SR3.2]	11							
[T3.4]	8										
[T3.5]	4										

LEGEND:		
	ACTIVITY	112 BSIMM6 activities, shown in 4 domains and 12 practices
	BSIMM6 FIRMS	count of firms (out of 78) observed performing each activity
		most common activity within a practice
		most common activity not observed in this assessment
	1	most common activity was observed in this assessment
		a practice where firm's high-water mark score is below the BSIMM6 average

Figure 1. The Building Security In Maturity Model 6 (BSIMM6) scorecard can be used to rate a target firm against the BSIMM population on II2 activities. To read about particular activities, download the BSIMM at http://bsimm.com.

YOUR FIRM NEEDS A SOFTWARE SECURITY GROUP

Each of the 78 software security initiatives described in BSIMM6 has a software security group (SSG). Successfully carrying out the BSIMM activities without an SSG is very unlikely (and hasn't been observed in the field to date), so it's essential to create an SSG before you start working to adopt the BSIMM activities.

SSGs come in a variety of shapes and sizes. Strong SSGs tend to include people with deep coding experience and architectural chops. Software security can't only be about finding specific bugs such as the Open Web Application Security Project Top 10 (www.owasp.org/index.php/Category:OWASP_Top_Ten_Project). Code review is a very important best practice, but reviewers must actually understand code (not to mention the huge piles of security bugs). However, the best code reviewers sometimes make very poor software architects, and asking them to perform an architecture risk analysis will only result in blank stares. Make sure code and architectural capabilities are equally covered in your SSG.

An SSG is often asked to mentor, train, and work directly with hundreds of developers. Communication skills,

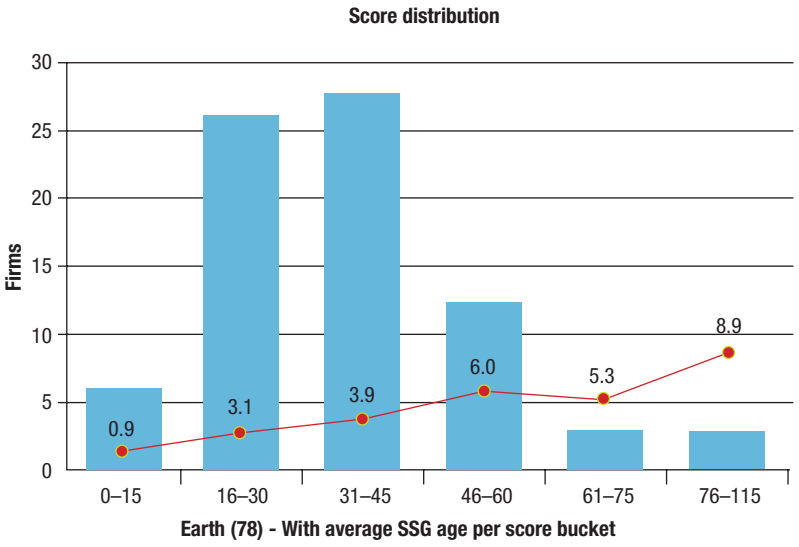


Figure 2. Distribution of BSIMM maturity scores among 78 firms (referred to as "Earth" in the BSIMM). SSG is software security group.

teaching capability, and good consulting sense are must-haves for at least a portion of the SSG staff. For more about this, see SearchSecurity's article "How to Build a Team for Software Security Management," which was based on SSG structure data gathered at the 2014 BSIMM Community Conference (http://searchsecurity.techtarget.com/opinion/McGraw-How-to-build-a-team-for-software-security-management).

Though no two of the 78 firms we examined had exactly the same SSG structure—suggesting that there are multiple ways to structure an SSG—we did observe some commonalities. At the highest level of organization, SSGs have five major roles:

- provide software security services,
- set policy,
- mirror business unit organizations,
- use a hybrid policy and services approach,
- and manage a distributed network of those doing software security work.

Some SSGs are highly distributed across a firm and others are very centralized. Looking at all the SSGs in our study, we see several common "subgroups": people dedicated to policy, strategy, and metrics; internal services groups that (often separately) cover tools, penetration testing, middleware development, and shepherding; incident response groups; training development and delivery groups; externally facing marketing and communications groups; and vendor-control groups.

We observed an average ratio of SSG to development of 1.51 percent across the entire group of organizations, meaning there's one SSG member for every 75 developers when we average the ratios for each participating firm. The largest ratio found was 16.7 percent and the smallest, 0.03 percent. The average SSG size among the 78 firms is approximately 14 people (range = 1–130, median = 6).

If you intend to take on software security in a firm-wide fashion, start by forming an SSG that's the right size to get the job done.

EXPERIENCE AND MATURITY MAKE A BIG DIFFERENCE

With a larger BSIMM data set than ever before, we can now analyze large-scale trends. For example, we were able to graph the distribution of maturity scores among the participating firms by dividing the scores into six bins (see Figure 2). The scores represent a slightly skewed bell curve. We also plotted the firms' average age in each bin, represented by the orange line on the graph. In general, firms with more observed BSIMM activities have older software security initiatives.

We also compared groups of firms by maturity. On average, the top 11 firms in the BSIMM population have a development group size of 10,000, have been doing software security at the enterprise level for 6.8 years, have an SSG with 29 members, and have a satellite (developers, architects, and others who are directly engaged in software security but aren't part of the SSG) of 118 people. In contrast, the bottom 11 firms have an average development group size of 600, have been doing software security at the enterprise level for 1.25 years, have an SSG with 3.4 members, and don't have a satellite.

Comparing activities commonly found among the top 11 versus bottom 11 firms is telling. Although six of the same activities are found in both populations (ranging from code review and training activities to data classification and standards activities), nine are observed in the top firms and none are observed in the bottom firms. These nine activities emphasize governance and outreach:

- SM1.1: Publish process (roles, responsibilities, plan); evolve as necessary.
- SM1.3: Educate executives.
- SM2.1: Publish data about software security internally.
- SM2.2: Enforce gates with measurements and track exceptions.
- CP1.3: Create policy.
- CP2.5: Ensure executive awareness of compliance and privacy obligations.
- SR1.2: Create a security portal.
- AM1.3: Identify potential attackers.

› AA1.4: Use a risk questionnaire to rank applications.

With this analysis, not only do we know the kinds of activities undertaken in more mature software security initiatives, but we also generally know when those activities are undertaken in the initiative's life cycle. Less mature firms, or those just getting started with software security, have plenty to learn from their more experienced peers.

We know what to do for software security and even how and when to do it. Now we just need to make it so everywhere.

SOFTWARE SECURITY SHOULD BE EVENLY DISTRIBUTED

One of the most commonly held myths of software security is that developers and development staff should just "take care of" software security. The theory is that with some training, developers can do it all. Our work with the BSIMM shows that this isn't the case, and that an SSG is necessary.

However, development staff and other members of a firm should eventually be directly involved in software security. In fact, satellites play a major role in executing software security activities among the most mature BSIMM community firms. BSIMM6 describes the work of 1,084 SSG members working directly with a satellite of 2,111 people (that's right—the satellite population is twice as large as the SSG population).

A satellite can be widely distributed with one or two members in each product group, or it can be more focused, getting together regularly to compare notes, learn new technologies, and expand the understanding of software security in an organization. Identifying and fostering a strong satellite is important to the success of many software security initiatives, but not all of them. Some BSIMM activities target the satellite explicitly.

Each of the 10 firms with the highest BSIMM scores has a satellite (100 percent) with an average size of 131 people. Thirty of the remaining 68 firms have a satellite (44.1 percent), and none of the 10 firms with the lowest BSIMM scores has a satellite. This suggests that as a software security initiative matures, its activities become distributed and institutionalized into the organizational structure. Among the BSIMM population of 78 firms, initiatives tend to evolve from centralized and specialized to decentralized and distributed, with an SSG orchestrating things at the core.

The time has come to put away the bug parade boogeyman (www.informit.com/articles/article.aspx?p=1248057), the top 10 tea leaves, the black box Web app goat sacrifice, and the occult reading of penetration testing entrails. It's science time. The BSIMM provides an important step forward in the institutionalization of software security as a discipline. Improvement is only possible when measurement is in place, and the BSIMM remains the only measurement tool in the software security field.

This article originally appeared in Computer, *vol. 49, no. 1, 2016.*

GARY MCGRAW, PhD, is Cigital's chief technology officer. Contact him at gem@cigital.com.

EDITORS: Rick Kuhn, US National Institute of Standards and Technology, kuhn@nist.gov
Tim Weil, US Department of Interior, trweil@ieee.org

Biological Warfare
Tampering With Implantable Medical Devices

Jay Liebowitz, *Harrisburg University of Science and Technology*
Robert Schaller, *University of Pennsylvania*

Almost everyone has seen a James Bond movie and has enjoyed all the intricate contraptions designed for 007. In today's throes of mystery and intrigue, we might even envision a modern-day hacker tinkering with an influential, high-ranking official's pacemaker to cause his or her ultimate demise. Research now seems to indicate that security concerns about implantable medical devices (IMDs) are not "Bond-ish"—they are real and potentially deadly.[1,2] Even back in 2007, Vice President Dick Cheney's cardiologist disabled the wireless functionality of his pacemaker because of just that fear.[3]

IMDs have been around since 1958. They include pacemakers, implantable cardiac defibrillators, insulin pumps, cochlear implants, and neurostimulators. In the US, they're regulated by the Food and Drug Administration (FDA). However, over the years, there has been a lack of focus on issues surrounding possible cyberattacks that tamper with IMDs.[2] We certainly have made advances in IMDs over the past six decades, especially in the areas of biocompabitility, structural design of devices and delivery systems, power management, and detection or wireless communication issues.[4] We still need, however, to take a closer look at security issues relating to IMDs in the near future.

Current Trends

According to a new market research report,[5] the nanotechnology in the medical devices market was valued at around US$5 billion in 2014 and expected to reach around $8.5 billion by 2019. Every year, about 300,000 Americans receive wireless medical devices, including IMDs, such as pacemakers, glucose monitors, and pain pumps.[6] The key question is, "Can these Internet-connected medical implants, like pacemakers, be hacked?" A new report in the journal *Science* suggests that this could be the case.[3] Many of these devices connect with a hand-held controller over short distances using Bluetooth.[3] According to the April 2015 issue of *Communications of the ACM*,

> Security and safety issues in the medical domain take many different forms. Examples range from purposely contaminated medicine to recalls of vascular stents, and health data breaches. Security risks resulting from intentional threats have only recently been confirmed, as medical devices increasingly use newer technologies such as wireless communication and Internet access. Intentional threats include unauthorized access of a medical device or unauthorized change of settings of such a device.[7]

In the coming years, more IMDs will be approved and used by patients. On 14 January 2015, the FDA approved the Maestro Rechargeable System, the first FDA-approved obesity device since 2007. According to the Centers for Disease Control and Prevention (CDC), one third of all US adults are obese (www.fda.gov/NewsEvents/Newsroom/PressAnnouncements/ucm430223.htm), suggesting a potentially broad user base.

Background

Cardiac implantable electronic devices (permanent pacemakers and

 Published by the IEEE Computer Society

implantable cardioverter defibrillators) are particularly appealing targets for cyberattacks due to the life-saving services that they provide and the complexity of their platforms. These devices utilize embedded software to monitor heart rates in patients with heart rhythm disorders. Randomized clinical trials have consistently shown benefits in certain patient populations, leading to a dramatic rise in implantation rates.[8]

Practitioners use commercial device programmers that communicate with these devices to extract clinical information, test particular features, and modify programming. Radio software makes it possible to communicate when in close proximity but without a direct connection. Recent advancements have allowed communication over the Internet to facilitate more expedient and efficient patient–physician communication, which has also resulted in financial savings. These same conveniences also make these devices more vulnerable to cyberattacks.

According to one study,[9] secure system design for IMDs has the following challenges:

- *Battery life.* Security algorithms increase IMDs' operation complexity and degrade the battery life dramatically.
- *Adaptability.* New countermeasures might be needed to combat new attacks, thus the IMD must be modifiable.
- *Availability.* IMDs should be available to a doctor for emergency treatment, even if that doctor wasn't previously authorized.
- *Reliability.* Security mechanisms should be robust enough to ensure system reliability.

Some security threats to IMDs fall into the categories of eavesdropping, impersonation, and jamming. Current approaches for dealing with these potential threats are through cryptography, external device deployment (pairing an external wearable device with the IMD), anomaly detection, and frequency hopping-spread spectrum (FHSS) and direct sequence-spread spectrum (DSSS) defenses.[9]

Is There Still a Concern?

So, the key question remains, "Can these IMDs be tampered with and hacked by others who aren't supposed to have access?" In 2008, William Maisel, a Harvard Medical School cardiologist, coauthored a paper that indicated that hackers could reprogram an IMD without authorization.[1] Using a commercial ICD programmer and a software radio, researchers were able to gain wireless access to a Medtronic defibrillator to garner personal patient data and enable potentially malicious therapy. Despite significant advantages (a team of researchers, expensive lab equipment, and close proximity to the device), they were able to show that potential security problems do exist.

The report encouraged IMD manufacturers to heed this warning and provide proper security precautions. Medtronic, Boston Scientific, and St. Jude Medical are the main manufacturers of defibrillators. Back in 2008, Medtronic (after seeing the report) said it was increasing the sophistication of the devices to make it difficult for outside tampering. Also, Boston Scientific said at that time that it used encryption in its defibrillators, and doubted its devices could be hacked.[1]

In 2013, Billy Rios and Terry McCorkle of Cylance reported a hard-coded password vulnerability affecting roughly 300 medical devices across approximately 40 vendors. Based on their research, the vulnerability could be exploited to potentially change critical settings or modify device firmware (https://ics-cert.us-cert.gov/alerts/ICS-ALERT-13-164-01).

A 2014 medical device cybersecurity workshop hosted by the FDA noted that "it's only a matter of time before a patient is killed or injured due to a targeted cyberattack against a medical device—or even as the result of an unintentional cyber vulnerability."[10]

Also, in 2014, the US Department of Homeland Security indicated that it was investigating about two dozen cases of suspected cybersecurity flaws in medical devices and hospital equipment that officials fear could be exploited by hackers.[11]

But while preemptive investigation in an effort to identify vulnerable firmware is one thing, criminal investigation after the fact is another. The medical examiner's office does not necessarily have the expertise or resources to perform basic device interrogations, let alone the capability to conduct composite computer forensics investigations.

What Still Needs to Be Done?

It's crucial to anticipate threats and create systems that are adequately protected. In addition to encouraging the industry to increase security, Maisel's team also presented "zero-power" defenses (that is, defenses that don't rely on the IMD's battery but rather harvest power from external RF), including audible notifications warning a patient of tampering or requiring that an incoming signal be authenticated. Additional research aimed at defending such activity involved the creation of a "noise shield" that can block out certain attacks, the use of ultrasound waves to determine the distance between a transmitter and a

medical device to prevent remote attacks, and the development of a biometric heartbeat sensor to allow devices within a body to communicate with each other, keeping out external devices and signals.[12–14] As with all evolving technologies, complex security systems must be balanced with ease of use for practitioners on a daily basis.

From an IT security perspective, this is certainly a new potential type of "biological warfare." IMD manufacturers must find ways to better secure their devices from possible outside hacking by incorporating security technologies such as encryption, and the FDA needs to study this issue carefully. If not, people may certainly be able to get at "the heartbeat of America"!

References

1. D. Halperin et al., "Security and Privacy for Implantable Medical Devices," *IEEE Pervasive Computing*, vol. 7, no. 1, 2008, pp. 30–39.
2. S. Gupta, "Implantable Medical Devices: Cyber Risks and Mitigation Approaches," *NIST Cyber Physical Systems Workshop*, 2012; http://csrc.nist.gov/news_events/cps-workshop/slides/presentation-1_gupta.pdf.
3. R. McDonough, "Security Concerns over Hacking of Pacemakers, Other Implanted Medical Devices," CBS, 12 Feb. 2015; http://philadelphia.cbslocal.com/2015/02/12/security-concerns-over-hacking-of-pacemakers-other-implanted-medical-devices/.
4. Y.H. Joung, "Development of Implantable Medical Devices: From an Engineering Perspective," *Int'l Neurourology J.*, vol. 17, no. 3, 2013, pp. 98–106.
5. *Nanotechnology in Medical Devices Market by Product Research Report*, Markets and Markets, 18 March 2015; www.marketsandmarkets.com/Market-Reports/nanotechnology-medical-device-market-65048077.html.
6. M. Goodman, "Implantable Medical Devices Hacking: Who Does the Autopsy?" *Slate*, 13 Mar. 2015.
7. J. Sametinger et al., "Security Challenges for Medical Devices," *Comm. ACM*, vol. 58, no. 4, 2015, pp. 74–82.
8. C.M. Tracy et al., "2012 ACCF/AHA/HRS Focused Update of the 2008 Guidelines for Device-Based Therapy of Cardiac Rhythm Abnormalities," *J. Am. College of Cardiology*, vol. 60, no. 14, 2012, pp. 1297–1313.
9. K. Ankarali et al., "A Comparative Review on the Security Research for Wireless Implantable Medical Devices," Academia.edu, 2015; www.academia.edu/10084615/A_Comparative_Review_on_the_Security_Research_for_Wireless_Implantable_Medical_Devices.
10. M. McGee, "Medical Device Hacks: The Dangers," Healthcare Info Security, 21 Oct. 2014; www.healthcareinfosecurity.com/medical-device-hacks-dangers-a-7464/op-1.
11. J. Finkle, "US Government Probes Medical Devices for Possible Cybersecurity Flaws," *Reuters*, 22 Oct. 2014; www.reuters.com/article/2014/10/22/us-cybersecurity-medicaldevices-insight-idUSKCN0IB0DQ20141022.
12. S. Gollakota et al., "They Can Hear Your Heartbeats: Non-Invasive Security for Implantable Medical Devices," *Proc. ACM SIGCOMM 2011 Conf.*, 2011, pp. pp. 2–13.
13. "Keeping Pacemakers Safe from Hackers," *MIT Technology Rev.*, 10 Nov. 2009.
14. C. Cornelius, "Who Wears Me? Bioimpedance as a Passive Biometric," *Proc. 3rd Usenix Conf. Health Security and Privacy*, 2012, pp. 4–4.

Jay Liebowitz *is the Distinguished Chair of Applied Business and Finance at Harrisburg University of Science and Technology. He previously was the Orkand Endowed Chair in Management and Technology at the University of Maryland University College, and full professor at Johns Hopkins University. Contact him at jliebowitz@harrisburgu.edu.*

Robert Schaller *is an assistant professor of clinical medicine at the Hospital of the University of Pennsylvania. He specializes in the treatment of heart rhythm disorders, including complex ablation therapy and implantation, and extraction of cardiac implantable electronic devices. Contact him at robert.schaller@uphs.upenn.edu.*

This article originally appeared in IT Professional, *vol. 17, no. 5, 2015.*

Managing Risk in a Cloud Ecosystem

Michaela Iorga
National Institute of Standards and Technology

Anil Karmel
C2 Labs

DUE TO ECONOMIES OF SCALE, CUTTING-EDGE TECHNOLOGY ADVANCEMENTS, AND HIGHER CONCENTRATION OF EXPERTISE, CLOUD PROVIDERS HAVE THE POTENTIAL TO OFFER STATE-OF-THE-ART CLOUD ECOSYSTEMS THAT ARE RESILIENT, SELF-REGENERATING, AND SECURE—FAR MORE SECURE THAN THE ENVIRONMENTS OF CONSUMERS WHO MANAGE THEIR OWN SYSTEMS. This has the potential to greatly benefit many organizations. The key to successful implementation of a cloud-based information system is a level of transparency into the cloud provider's service. This level of transparency allows businesses to build the necessary trust and to properly weigh the benefits of adopting such solutions. In this assessment process, businesses need to consider the sensitivity of the stored information against the incurred security and privacy risks. For example, the benefits of a cloud-based solution would depend on the cloud model, type of cloud service considered, type of data involved, the system's criticality/impact level, cost savings, service type, and any associated regulatory requirements.

Cloud-based information systems are exposed to threats that can have adverse effects on organizational operations (such as missions, functions, image, or reputation), organizational assets, individuals, and other organizations. Malicious entities can exploit both known and unknown vulnerabilities to compromise the confidentiality, integrity, or availability of the information being processed, stored, or transmitted by those systems.

Risk management activities can be grouped based upon the level at which they address the risk-related concerns:

- organization level (tier 1);
- mission and business process level (tier 2); and
- information system level (tier 3).

In this article, we focus only on the tier 3 security risks related to the operation and use of cloud-based information systems. To prevent and mitigate any risks, adverse actions, service disruptions, attacks, or compromises, organizations need to quantify their residual risk (that is, the portion of risk remaining after security measures have been applied[1]) below the threshold of the acceptable level of risk.

Security Risk and Cloud

Information systems risk management (tier 3) is guided by the risk decisions at tier 1 and tier 2. Information security requirements are satisfied by the selection of appropriate management, operational, and technical security controls from standardized catalogs of security and controls.[2–4]

Volume 1 of National Institute of Standards and Technology (NIST) Special Publication (SP) 500-293, *US Government Cloud Computing Roadmap*, highlights that boundaries in a cloud ecosystem are more complex and therefore renders traditional risk management mechanisms, such as perimeter-based defense mechanisms, less effective.[5] Moreover, in a cloud ecosystem, the complex relationships among cloud actors,[6]

2376-113X/16/$33.00 © 2016 IEEE Published by the IEEE Computer Society March 2016

the actors' individual missions, business processes, and their supporting information systems require an integrated, ecosystem-wide risk management framework (RMF) that addresses all cloud actors' needs. As with any information system, for a cloud-based information system, cloud actors are responsible for evaluating their acceptable risk, which depends on the threshold set by their risk tolerance to the cloud ecosystem-wide residual risk.

In general, organizations have maximum flexibility in how risk assessments are conducted. Because risk assessments facilitate decision making at all three tiers (organization level, mission/business process level, and information system level), they're key processes of effective risk management and in maintaining the residual risk below the threshold, and therefore the methods employed to assess the risks are of crucial importance. We recommend reading NIST SP 800-30, *Guide for Conducting Risk Assessment*, which provides quantitative, qualitative, or semiqualitative methods that use scores or levels, respectively.[7]

To effectively manage information security risk at the ecosystem level, the following high-level elements must be established:

- Assignment of risk management responsibilities to the cloud actors involved in the orchestration of the cloud ecosystem. Internally, cloud actors need to further assign responsibilities to their senior leaders, executives, and representatives.
- Establishment of a cloud ecosystem-wide tolerance for risk and communication of this risk tolerance through service-level agreements (SLA), including information on decision-making activities that impact the risk tolerance.
- Near real-time monitoring, recognition, and understanding, by each cloud actor, of the information security risks arising from the operation and/or use of the information system leveraging the cloud ecosystem.
- Accountability by the cloud actors and near real-time information sharing of the cloud actors' incidents, threats, risk management decisions, and solutions.

Risk is often expressed as a function of the *magnitude of harm* caused by the occurrence of a circumstance or event, multiplied by the *likelihood of* its *occurrence*. In information security, *likelihood of occurrence* is a weighted risk factor based on an analysis of the probability that a given *threat* is capable of exploiting a given *vulnerability*. Accordingly, security risk assessments focus on identifying where in the cloud ecosystem damaging events could take place.

The risk-based approach to managing information systems is a holistic activity that needs to be fully integrated into every aspect of the organization. An RMF provides a disciplined and structured process that integrates information security and risk management activities into the system development life cycle. An RMF operates primarily at tier 3 in the risk management hierarchy, but it can also have interactions at tier 1 and tier 2.

NIST SP 800-37, Revision 1, *Guide for Applying the Risk Management Framework to Federal Information Systems*, introduces a risk management process mandated for federal agencies but widely vetted by state and local governments and by private sector organizations as a best practice for traditional information systems.[8] As that document states, defining information system requirements is a critical part of any system development process and needs to begin in a system's initiation phase. Since the security requirements are a subset of the overall functional and nonfunctional requirements, security requirements need to be integrated into the system development life cycle (SDLC) simultaneously with the functional and nonfunctional requirements. Treating security as a patch or addition to the system and architecting and implementing solutions independent of the SDLC is a more difficult process that can incur higher costs with a lower potential to effectively mitigate risk.

We encourage you to review NIST SP 800-37, Revision 1, as well, which we use here as a reference framework for the current discussion of applying the RMF in a cloud ecosystem. For the sake of brevity, we won't review in this article the six steps and the tasks described in that document. It's important to note that even though the NIST document addresses complex information systems composed of multiple subsystems operated by different entities, it doesn't address cloud-based information systems, or any other kind of systems that leverage utility-based resources, and hence the need for the current discussion.

When orchestrating a cloud ecosystem for a cloud-based information system, cloud consumers, as owners of the data associated with the system, remain responsible for securing the system and the data commensurate with the data sensitivity. However, cloud consumers' level of control and direct management varies based on the cloud deployment model. NIST defined in SP 800-145, *The NIST Definition of Cloud Computing,* the cloud, cloud deployment models (public, private, hybrid, and community), and cloud service models (infrastructure as a service [IaaS], platform as a service [PaaS], and software as a service [SaaS]).[9] In an IaaS cloud, the cloud consumer manages the top part of the functional stack above the hypervisor, while the consumer-managed functional stack proportionally decreases for a PaaS cloud and is reduced to a minimum in a SaaS cloud ecosystem.

The RMF introduced in NIST SP 800-37, Revision 1 is applicable by a cloud actor to the layers of the functional stack that are under management. In a simplified cloud ecosystem model, which is orchestrated only by the cloud consumer and the cloud provider, the cloud provider applies the RMF to the lower part of the stack, which is built as part of the service offered. Cloud consumers will apply the RMF to the upper functional layers, the ones built and deployed on top of the cloud infrastructure offered as a service.

However, prior to acquiring a cloud service, a cloud consumer needs to analyze the risk associated with adopting a cloud-based solution for a particular information system, and plan for the risk-treatment and risk-control activities associated with the cloud-based operations of this system. To do so, a cloud consumer needs to gain the perspective of the entire cloud ecosystem that will serve the operations of their cloud-based information system. Cloud consumers must also apply the RMF in a customized way that allows them to

- perform a risk assessment,
- identify the best-fitting cloud architecture,
- select the most suitable cloud service,
- gain necessary visibility into the cloud offering, and
- define and negotiate necessary risk treatment and risk control mitigations before finalizing the SLA and proceeding with the security authorization.

Figure 1 depicts this RMF for the cloud ecosystem (RMF4CE) from the cloud consumer's perspective, showing it as a repeatable process that encompasses the entire cloud ecosystem.

In a cloud ecosystem, cloud consumers must establish the clear demarcation of information-system boundaries on all levels in a vendor-neutral manner. Furthermore, the cloud consumer must establish measures to ensure appropriate protection, regardless of vendor, ownership, or service level for the cloud-based information system.

Cloud Provider's Risk Management Process

A cloud provider's selection and implementation of its security and privacy controls consider their effectiveness, efficiency, and constraints based on the applicable laws, directives, policies, standards, or regulations with which the provider must comply. The cloud consumers' specific requirements and mandates are unknown and therefore are projected as a generic core set.

Cloud providers have significant flexibility in determining what constitutes a cloud service and therefore its associated boundary, but at the time the system is architected and implemented, they can only assume the nature of data their cloud consumers will generate. Therefore, the security and privacy controls selected and implemented by a cloud provider are sets that meet the needs of a large number of potential consumers. However, the centralized nature of the offered cloud service enables a cloud provider to engineer highly technical, specialized security solutions that can provide a higher security posture than that in traditional IT systems.

Applying standardized or well-vetted approaches to cloud service risk management is critical to the success of the entire cloud ecosystem and its supported information systems. Since the offered cloud service is directly managed and controlled by the cloud provider, applying the RMF to this system doesn't require additional tasks beyond those of a classical IT system; therefore, a risk management approach like the one discussed previously is a good example of a broadly accepted, well-vetted approach.

It's important to note that a cloud ecosystem's security posture is only as strong as the weakest subsystem or functional layer. Since a cloud provider's reputation and business continuity depend

FIGURE 1. Applying a risk management framework (RMF) to a cloud ecosystem (RMF4CE). (Functional stack image courtesy of Cloud Security Alliance, 2009)

on the smooth operation and high performance of their consumers' solutions, when applying the RMF a cloud provider aims to compensate for possible weakness in their cloud consumers' solutions.

Cloud Consumer's Risk Management Process

For successful adoption of a cloud-based information system solution, the cloud consumer must be able to clearly understand the system's cloud-specific characteristics, the architectural components for each service type and deployment model, and the cloud actors' roles in establishing a secure cloud ecosystem. Furthermore, it is essential to cloud consumers' business and mission-critical processes that they have the ability to

- identify all cloud-specific, risk-adjusted security and privacy controls;
- request from the cloud providers and brokers (when applicable and via contractual means) service agreements and SLAs where the cloud providers are responsible for implementing security and privacy controls;
- assess the implementation of said security and privacy controls; and
- continuously monitor all identified security and privacy controls.

Since the cloud consumers directly manage and control the functional capabilities they implement, applying the RMF to these functional layers doesn't

require more tasks or operations than necessary in a classical IT system; therefore, the risk management approach discussed earlier is a good example of a broadly accepted, well-vetted approach.

With cloud-based services, some subsystems or subsystem components fall outside the direct control of a cloud consumer's organization. Since a cloud-based solution doesn't inherently provide the same level of security and compliance as the traditional IT model, being able to perform a comprehensive risk assessment is key to building trust in the cloud-based system as the first step in authorizing its operation.

Cloud characteristics often present a cloud consumer with security risks that are different from those in traditional information technology solutions. To preserve the security level of their information system and data in a cloud-based solution, cloud consumers must be able to identify all cloud-specific, risk-adjusted security and privacy controls in advance of cloud service acquisition. They must also request from the cloud providers and brokers, through contractual means and SLAs, that all security and privacy components are identified and that their controls are fully and accurately implemented.

Understanding the relationships and interdependencies between the different cloud computing deployment models and service models is critical to understanding the security risks involved in cloud computing. The differences in methods and responsibilities for securing different combinations of service and deployment models present a significant challenge for cloud consumers. They need to perform a thorough risk assessment to accurately identify the security and privacy controls necessary to preserve their environment's security level as part of the risk treatment process, and to monitor the operations and data after migrating to the cloud in response to their risk control needs.

In general, a cloud consumer adopting a cloud-based solution needs to follow the same RMF steps discussed earlier in addition to the tasks listed in Table 1. The table aligns risk management activities with their corresponding steps from NIST SP 800-37, Revision 1, and provides additional tasks (in italics) that map to Figure 2.

The RMF applied to the cloud ecosystem from the consumer's perspective can be used to address the security risks associated with cloud-based information systems by incorporating the outcome into the terms and conditions of the contracts with external cloud providers and cloud brokers. Performance aspects of these terms and conditions are also incorporated into the SLA, which is an intrinsic part of the security authorization process and of service agreements between the cloud consumer, provider, and broker (when applicable). Contractual terms should include guarantees of the cloud consumer's timely access to or the provider's timely delivery of cloud audit logs, continuous monitoring logs, and any user access logs.

The approach covered by the steps in Table 1 enables organizations to systematically identify their common, hybrid, and system-specific security controls and other security requirements to procurement officials, cloud providers, carriers, and brokers.

BEFORE ADOPTING A CLOUD-BASED SOLUTION FOR AN INFORMATION SYSTEM, CLOUD CONSUMERS MUST DILIGENTLY IDENTIFY THEIR SECURITY REQUIREMENTS. In addition, they must assess each prospective service provider's security and privacy controls, negotiate SLAs and service agreements, and build trust with the cloud provider before authorizing the service. A thorough risk analysis coupled with the secure cloud ecosystem orchestration introduced here, along with adequate guidance on negotiating SLAs, are intended to assist cloud consumers in managing risk and making informed decisions when adopting cloud services.

References

1. *National Information Assurance (IA) Glossary*, Committee on National Security Systems Instruction No. 4009, Apr. 2010; www.ncsc.gov/nittf/docs/CNSSI-4009_National_Information_Assurance.pdf.
2. National Institute of Standards and Technology, *Security and Privacy Controls for Federal Information Systems and Organizations*, NIST Special Publication 800-53, Revision 4, 2013; http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r4.pdf.
3. International Organization for Standardization, *Information Technology—Security Techniques—Information Security Management—Requirements*,

Table 1. Risk management framework (RMF) applied to a cloud ecosystem from a cloud consumer's perspective.

Risk management activities	NIST SP 800-37 RMF steps	Description
Risk assessment (analyze cloud environment to identify potential vulnerabilities and shortcomings)	1. Categorize	Categorize the information system and the information processed, stored, and transmitted by that system based on a system impact analysis. Identify operational, performance, security, and privacy requirements.
	2. Select (includes evaluate-select-negotiate)	*Identify and select functional capabilities for the entire information system.* Identify and select the associated baseline security controls based upon the system's impact level, and the privacy controls. Tailor and supplement the security controls by selecting enhancements and/or additional controls deemed necessary.
		Identify and select best-fitting cloud architecture for this information system.
		Evaluate/review cloud providers that meet consumer's criteria (architecture, functional capabilities, and controls).
		Select cloud provider(s) that best meet(s) the desired architecture and the security requirements (ideally should select the provider that provides as many controls as possible to minimize the number of controls that will have to be tailored). *In the process, identify the controls that will be implemented by the consumer, the controls implemented by the provider as part of the offering, and the controls that need to be tailored (via compensating controls and/or parameter selection).*
		Negotiate SLA, metrics, and sign service agreement as part of the procurement process. Document all the controls in the security plan. Review and approve the security plan.
Risk treatment (design mitigation policies and plans)	3. Implement	Implement security and privacy controls for which the cloud consumer is responsible.
	4. Assess	*Assess the cloud provider's implementation of the tailored security and privacy controls.*
		Assess the implementation of the security and privacy controls, and identify any inheritance and dependency relationships between the provider's controls and consumer's controls.
	5. Authorize	Authorize the cloud-based information system to operate.
Risk control (risk monitoring—surveying, reviewing events, identifying policy adjustments)	6. Monitor	Continuous/near real-time monitoring of operations and effectiveness of the security and privacy controls under consumer's management. *Continuous/near real-time monitoring of cloud provider's operations related to the cloud-based information system and assessment of the systems' security posture.* *Reassess and reauthorize (periodic or ongoing) the cloud provider's service.*

ISO/IEC 27001, 2013; www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=54534.

4. International Organization for Standardization, *Information Technology—Security Techniques—Code of Practice for Information Security Controls*, ISO/IEC 27002, 2013; www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=54533.
5. L. Badger et al., *US Government Cloud Computing Technology Roadmap*, NIST Special Publication 500-293, volumes 1 and 2, 2014; http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.500-293.pdf.

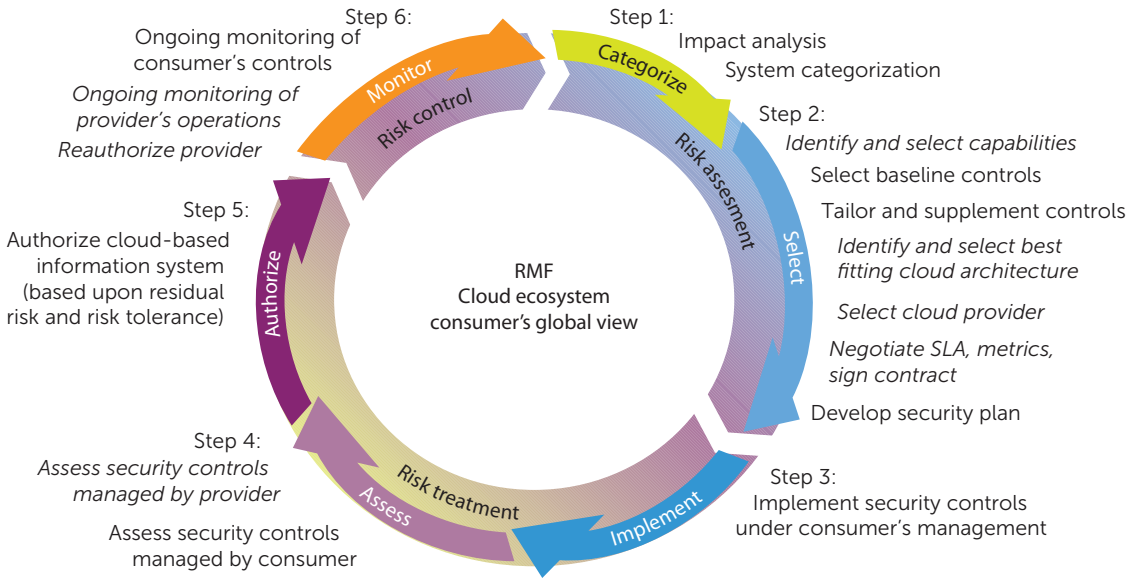


FIGURE 2. Cloud consumers' view of the risk management framework (RMF) applied to a cloud ecosystem.

6. F. Liu et al., *NIST Cloud Computing Reference Architecture*, NIST Special Publication 500-292, 2011; www.nist.gov/customcf/get_pdf.cfm?pub_id=909505.
7. National Institute of Standards and Technology, *Guide for Conducting Risk Assessment*, NIST Special Publication 800-30, Revision 1, 2012; http://csrc.nist.gov/publications/nistpubs/800-30-rev1/sp800_30_r1.pdf.
8. National Institute of Standards and Technology, *Guide for Applying the Risk Management Framework to Federal Information Systems: A Security Life Cycle Approach*, NIST Special Publication 800-37, Revision 1, 2010; http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-37r1.pdf.
9. P. Mell and T. Grance, *The NIST Definition of Cloud Computing*, NIST Special Publication 800-145, 2011; http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf.

MICHAELA IORGA *is senior security technical lead for cloud computing at the National Institute of Standards and Technology. Her current research interests include cloud computing security, forensics and privacy, information assurance, and federated identity and credential management issues in the cyberspace. Iorga has a PhD in engineering from Duke University. Contact her at michaela.iorga@nist.gov.*

ANIL KARMEL *is the cofounder and CEO of C2 Labs as well as the cochair of the National Institute of Standards and Technology's Cloud Security Working Group. His research interests include cloud computing security and privacy, secure DevOps, and container and microservices security. Karmel has a bachelor of science degree from the University of Illinois, Urbana/Champaign. Contact him at akarmel@c2labs.com.*

This article originally appeared in
IEEE Cloud Computing, *vol. 2, no. 6, 2015.*

EDITORS: Rick Kuhn, US National Institute of Standards and Technology, kuhn@nist.gov
Tim Weil, US Department of Interior, trweil@ieee.org

Protecting Digital Assets

Legal Protections ≠ Practical Security

Joseph Webster, Max Romanik, and Christopher Webster, *ShieldMyfiles*

The Internet is a lawless place. Before the digital era, items such as books, photographs, and mail were all owned in the physical world. Physical ownership comes with legal protections against theft, misappropriation, and seizure. As books become e-books, cameras become just another connected Web application, and communications become entirely digital, the law struggles to apply traditional legal concepts of privacy and property ownership to digitization. Here, we examine the ways in which traditional legal protections fail to meet the needs of digital property owners, looking first at two legal concepts relevant to protecting digital assets. We then examine how these laws apply—or don't—to data in the digital world.

Ownership

Legal ownership of property can be best described as a bundle of exclusive rights, including possession, use, and disposition.[1] Strong property rights exist when a single person or entity can substantiate that he or she has actual and exclusive possession of, is in use of, and is free to dispose of a given piece of property. The old adage "possession is nine-tenths of the law" does not mean that use and disposition are worth only one-tenth of a legal property claim. Rather, possession presents a rebuttable presumption that the possessor is indeed the lawful owner.[2] Take, for example, a Ferrari—the driver in possession is presumed to be the lawful owner.

The full bundle of rights is difficult to effectuate in our modern digital world. For example, in the case of the digital file "example.txt," the file's original creator is in exclusive possession, can use the file in any way he or she wishes, and is free to dispose of the file by sale, deletion, copying, sharing, and so on. This seems like a clear-cut example of property rights in practice. Digital assets, however, have certain characteristics that physical or analog property do not. File duplication is one characteristic that makes enforcement of classic property rights less certain. Duplication defeats exclusivity of possession because a copied file can be held by multiple users who can use or dispose of it in any way they wish. Contrast this digital reality with the physical realities of any piece of tangible property, like our Ferrari. The Ferrari cannot be copied for personal gain, stored all over the world, and seamlessly replaced when lost.

The Fourth Amendment

In addition to property law, digital assets can fall under the protections of the Fourth Amendment to the US Constitution, which provides protections from the US government. To fall under this protection, digital assets must be entitled to a "reasonable expectation of privacy."[3] More specifically, an owner must exhibit "an actual expectation" of privacy, and that expectation must be one "that society is prepared to accept."[4] Simply put, to have an expectation of privacy owners must take some steps to protect their property, and those steps must be ones that are recognized by society as valid means of protecting the privacy of that property (for example, storing documents in a locked safe).

 Published by the IEEE Computer Society

Data at Rest

It is difficult to apply the laws of property ownership and Fourth Amendment privacy protections to the digital world. The relationship between users and cloud services aptly illustrates these difficulties. This relationship is defined by the contract for services that is accepted when the user establishes an account. The service is a third party to whom users are voluntarily giving stewardship of some of their digital property. Contained in a typical set of terms are a number of provisions that are inconsistent with full property ownership. Google's terms of service state, "you retain ownership of any intellectual property rights that you hold in [your] content. In short, what belongs to you stays yours." But these terms also grant to Google a "worldwide license to use, host, store, reproduce, modify, create derivative works, communicate, publicly perform, publicly display, and distribute such content" (see www.google.com/intl/en/policies/terms/). The first clause creates and maintains the user's property rights, but the second is inconsistent with notions of sole ownership. While cloud providers such as Google would not be able to provide their products and services without this broad license, the result is that exclusive ownership is not held by the user. Rather, this is shared ownership in which users give up some portion of their rights. Additionally, cloud providers are not governmental entities. Thus, Fourth Amendment protections are not available to users, who have already consented to cloud providers viewing, using, hosting, storing, and modifying their files.

Worse yet, if the government wants to seize a user's files from a cloud provider, it can compel the provider to turn over files—in some cases without a warrant! The Stored Communications Act[5] presents two methods for the government to seize and search stored electronic records. If the records have been stored for less than 180 days, a warrant is required in all circumstances. If the records have been stored for 180 or more days, the government has three options: seize with a warrant, and demand that the user not be notified of the seizure indefinitely; seize with an administrative subpoena, and demand that the user not be notified of the seizure for 90 days; or seize with a subpoena, and within 90 days, file a warrant for the same records, providing the ability to extend the 90-day notice delay indefinitely.[5]

Contrast cloud data with property that is locked in a filing cabinet: the contents of the cabinet are free from prying eyes; the cabinet maker does not search the contents in return for providing the key; and if the government wishes to gain access to the cabinet, it must obtain a warrant based on probable cause. Another, more analogous example is a safety deposit box—the critical similarity being the hiring of a third party to act as a steward. Here, terms of service do not grant the bank license to use the contents of the box. Banks do not view, scan, copy, share, or transport any of the items in the box; the bank cannot inform on you about the contents of the box to intellectual property rights holders.

Data in Motion

The development of modern legal doctrine around the privacy of property in transit began with the US Postal Service (USPS). Since the nineteenth century, the US Supreme Court has accepted that envelopes and other packaging, which shield a parcel's contents from plain sight, create a reasonable expectation of privacy.[6] The only features of these forms of communication that are subject to inspection are the "outward form and weight."[6] In *Ex Parte Jackson*, the Court specifically pointed out that a distinction must be made between mail that is intentionally shielded from inspection, such as letters and sealed packages, and mail that is open to inspection, like magazines or post cards. The constitutional guaranty of privacy attaches to a person's property "where ever [it] may be," including while in transit with the USPS.[6] Mail in transit may only be opened and examined under a warrant "particularly describing the thing to be seized, as is required when papers are subjected to search in one's own household."[6] This law is not antiquated; rather, it forms the bedrock on which more modern recitations of the rule are built.[7]

Digital communications (that is, data moving over any common networking protocol) do not cleanly fit this constitutional standard because the Internet was built to be public by design. The USPS was designed to protect the privacy of sealed packages with locked mailboxes, secure sorting facilities, and its own police force. The Internet is replacing secure networks such as the USPS, even though many of the privacy and security mechanisms haven't been addressed. The incongruity between the public design of the Internet and the now private desires of its users creates a legal and regulatory gulf.

Communications routing information does not experience a reasonable expectation of privacy[8]; as such, physical addresses or dialed phone numbers are not private. Accordingly, many courts cast packet metadata in the same light. After all, how could information freely shared with the network and its participants enjoy any

expectation of privacy? Theoretically, this seems like a reasonable extension of the rule; however, it incorrectly analogizes packet metadata with phone numbers or address information. Packet metadata conveys more information than the simple "to" and "from" data that is found in addressing information. In aggregate, it can paint complex maps of interconnection and association that can intrusively reveal far more than simple addressing information about the parties involved. Furthermore, unlike mail sent through USPS, Internet traffic is not carried by delivery agents; it is designed to route packets publically through a flexible patchwork of nodes. Thus, the routing that makes the Internet possible comes at a privacy cost—it reveals packet data to every Internet routing participant.

The revelations by Edward Snowden regarding practices at the US National Security Agency (NSA) demonstrate how widespread bulk metadata collection programs have become. As authorization for the bulk collection of packet metadata, the NSA used the Uniting and Strengthening America by Providing Appropriate Tools Required to Intercept and Obstruct Terrorism Act (or "Patriot Act").[9] It is now expired, but was updated as the Uniting and Strengthening America by Fulfilling Rights and Ending Eavesdropping, Dragnet-collection and Online Monitoring Act (or "USA FREEDOM Act").[10] With its expiration, the challenges regarding the constitutionality of the Patriot Act are settled (likely candidates include *ACLU v. Clapper*, no. 13-3994, 2013; *Klayman v. Obama*, no. 13-0881, 2013; and *Smith v. Obama*, no. 14-35555, 2014). However, the legal precedent equating metadata with dialed phone numbers[8] remains unchallenged because the data is publicly known and required to route transmissions (follow the appeals of *ACLU v. Clapper*).

Additionally concerning is the fact that more than just packet metadata gets trapped, copied, and read by third parties. In fact, deep-packet inspection is an increasingly important network administration tool, used commonly in firewalls and data-loss-prevention systems. While packet capture might not be the product of malfeasance, it is inconsistent with the basic principles of property ownership and constitutional notions of privacy. Returning to the mailed letter analogy, there is an expectation, both practically and at law, that a mail carrier will not open a sealed envelope, make copies of the contents to read and store, and then reseal the envelope before delivery. In fact, such mail tampering would be a federal crime. Here again, we see the conflict between classical property rights and their application to the digital world.

Legal Protection ≠ Practical Protection

Legal protections do not cleanly apply to the digital world, and the woes of digital property owners do not end there. Frequently, a legal remedy provides no practical help. Take a legal victory in a criminal case: the remedy is exclusion of the seized and searched evidence at trial. In civil cases, the remedy is monetary compensation. Most digital property owners, however, are not looking for evidentiary exclusion or money; they want barriers to data loss and real privacy protections. Moreover, most violators of digital property rights are often not deterred by the law. If fact, you probably know someone who rampantly ignores digital property ownership.

There are two hurdles that digital property owners face in attempting to reform legal protections. The first is legislative change. To curtail government surveillance programs, the laws that govern the operations of surveillance and law enforcement agencies would have to be amended to restrict their activities or limit their funding. The passage of the USA FREEDOM Act shows that this option is time consuming, subject to political dispute, and delivers imperfect outcomes. The second hurdle is an individual's right of contract. Placing restrictions on terms-of-service agreements for third-party providers could infringe on the service providers' and users' ability to freely enter into contracts.[11] Finally, these reforms do nothing to deter cybercriminals who rampantly ignore the law entirely.

As we move to a more digital world that relies on cloud computing and greater connectedness, we are required to recognize the security vulnerabilities of digital property. This issue is both legal and technological by nature, and conforming current legal paradigms to digital property proves untenable. More proactive protective measures that enhance the security of digital property where traditional legal protections fall short are needed. The legal landscape is ever-changing, and it is likely that a proper legal paradigm will eventually emerge. Until that day comes, however, users should take proactive steps to protect themselves.

References

1. J. Wilson, "On the History of Property," 1804.
2. Ghen v. Rich, *Federal Reporter*, vol. 8, 1881, p. 159.
3. Katz v. United States, *US Reports*, vol. 389, 1967, p. 347 (Harlan concurring).

4. Katz v. United States, *US Reports*, vol. 389, 1967, p. 347.
5. US Code, Title 18, sections 2701–2712, 1986, as amended.
6. Ex Parte Jackson, *US Reports*, vol. 96, 1877, p. 727.
7. United States v. Choate, *Federal Reporter*, 2nd Series, vol. 576, 1978, p. 165 (US Court of Appeals for the 9th Circuit).
8. Smith v. Maryland, *US Reports*, vol. 442, 1979, p. 735.
9. Uniting and Strengthening America by Providing Appropriate Tools Required to Intercept and Obstruct Terrorism, Public Law No. 107-56, section 209, *US Statutes at Large*, vol. 115, 2001, p. 285.
10. Uniting and Strengthening America by Fulfilling Rights and Ending Eavesdropping, Dragnet-collection and Online Monitoring Act, Public Law No. 114-23, *US Statutes at Large*, vol. 129, 2015, p. 267.
11. *Restatement (Second) of Contracts*, section 187, 1981.

Joseph Webster *is a founder at ShieldMyfiles and a software and systems security architect. He specializes in high-quality secure software systems to protect privacy, valuable assets, and intellectual property. Webster is a passionate advocate for privacy and secure development and testing methodologies to promote best security practices for all. He is a Certified Information Systems Security Professional (CISSP). Contact him at joe@shieldmyfiles.com, or via https://www.linkedin.com/in/josephwebster.*

Max Romanik *is a founder at ShieldMyfiles. His research interests include the intersection of law, technology, and privacy, as well as healthcare technology and security. Romanik received a JD from the University of Maryland School of Law and an MBA from the Robert H. Smith School of Business at the University of Maryland. He is a member of the Maryland State Bar and is a professional member of IEEE. Contact him at max@shieldmyfiles.com or on Twitter @MaxRomanik.*

Christopher Webster *is a founder at ShieldMyfiles. His research interests include digital privacy law and policy, emergency preparedness, and crisis management. Webster received a JD from the University of Maryland School of Law. He is a member of the Maryland State Bar and is a professional member of IEEE. Contact him at chris@shieldmyfiles.com or on Twitter @christophersw1.*

This article originally appeared in IT Professional, *vol. 17, no. 6, 2015.*

The Research Horizon: Four Nearly Practical Concepts

Hilarie Orman • *Purple Streak*

Does the world of academic research have anything to contribute to the world of practical security? That was the question I asked myself recently while attending the annual IEEE Symposium on Security and Privacy. This year there were 55 papers, a record number. At least one of them was considered practical, because it won the "Distinguished Practical Paper" award. I found at least three other papers that seemed both practical and interesting: ways of using a hybrid capability architecture for a variety of process protection mechanisms, a near-term solution to privacy of DNS queries, and a methodology that turned up several exploitable errors in Transport Layer Security (TLS) state machine implementations. That last paper was one of two selected for the overall "Distinguished Paper" award.

In some sense, almost all security research papers are practical. They must surmount a high bar by covering new ground. Solutions that haven't been tried before, attacks that haven't been analyzed, and new ways to analyze existing systems are all aspects of the collection of papers for a good security research conference.

Today's research papers usually address a real-world problem in some way. The researchers will apply analysis to gain a deeper understanding of the problem, or they'll investigate a proposed solution's effectiveness. Some solutions aren't practical due to computational overhead or the amount of infrastructure that would have to be changed, but generally the paper is about something that might be used in the foreseeable future. Nonetheless, some things are more practical than others, and some things are more important than others. The four papers I have selected to discuss in this issue's "Practical Security" column occupy different places on the axes of practicality, novelty, and importance.

Practically Competing for Attention

The first paper is "Ad Injection at Scale: Assessing Deceptive Advertisement Modifications," and it's an investigation into how some ads get placed on webpages.[1] Webpages are a digital battleground when it comes to ads. There's a complex ecosystem feeding that war, and part of the system relies on some shady characters who use browser extensions to turn the ad selection system upside down.

When visiting a retailer's webpage, a user expects to see ads for products sold through the site. Similarly, when visiting a search results page, the user understands that some of the entries are paid for by advertisers. The retailers might think they're paying for an ideal kind of tailored shopping experience — a thoughtful selection of products suited to the shopper's preferences and budget and current best deals offered by the retailer. This is unlikely to be the experience for users who have installed browser extensions that promise an enhanced shopping experience.

A user might visit Walmart's website looking for a TV and see ads for Crazy Ed's Electronics, as a hypothetical example. Those ads weren't there when the Walmart server composed the page, but by the time the page is displayed in an afflicted user's browser, the entire process of selecting and inserting ads has been repeated by code that was installed as a browser extension. The extensions often are touted as advantageous for the user. In theory, they automatically do product searches and price comparisons. In practice, they're a nuisance because they slow down all page loads and don't deliver the promised value. On top of that, they can be almost impossible to remove.

The researchers were able to learn a great deal about the prevalence of ad injectors in the wild by utilizing their association with Google. Their methodology inserted monitoring code

 Published by the IEEE Computer Society

into webpages delivered by Google servers, but the code did nothing to change what was displayed. Any ads that were inserted after the page was delivered were undisturbed. The monitoring software waited for the "unload" event generated when the user navigated away from the page, and at that point the page's contents were compared to what was originally delivered. In this way, they learned that a few percent of all webpage deliveries are affected by ad injectors. The injectors affect many kinds of browsers, notably Chrome, Firefox, and Internet Explorer.

The research project then moved on to investigating the variety of software libraries that implement ad injection, the software providers, and the revenue stream that supports the practice. Unsurprisingly, only a few entities dominate ad replacement technology: Superfish, JollyWallet, VisAdd, and Nav-Links. Superfish is notorious because of its short-lived deal with Lenovo to preinstall its ad software on Windows machines, thus opening a security hole.[2] The researchers found that the advertisers who work with the ad injection companies form a tangled web, sometimes looping back to the unknowing retailer, who can end up paying for his own ads to be replaced with different ads of his own!

An interesting side note on this subject is in a different paper from the IEEE Symposium on Security and Privacy, one that detects and analyzes the function of embedded scripts in webpages.[3] I wonder if the research behind it might have come across the monitoring scripts used in the ad injection research.

While you have to admire the depth of the investigation and interesting information about the danger of browser extensions, the work leaves me wondering, Where's the security, where's the practicality? Web browsers are applications with unlimited extensibility and no security model, and it seems obvious that they're vectors for unpleasant surprises. Is ad replacement a security problem, or is it really a tug-of-war between competing ad companies? Is the user harmed by the ad replacements? Could there be ad replacement services that are mutually beneficial to the user and the website owner? It's likely that the war for user attention is only beginning, and this article won't be the last word on the subject.

Practical Privacy for DNS Queries

Web browsing is critically dependent on DNS lookups, a little piece of magic that turns a name like Walmart.com into an Internet address. Although the authenticity of the DNS response can be assured through cryptography, and although the webpage that you ultimately access may be delivered with privacy-preserving encryption, the DNS query itself is in plain view of all network-observing eyes. Can this privacy gap be closed?

Researchers at Verisign Labs and USC's Information Sciences Institute teamed up to validate the assertion that it's eminently feasible and practical to use TLS to gain this privacy.[4] Their work shows that the fears of server overload can be allayed through careful connection management, a few implementation changes, and a protocol extension. Their conclusions are based on analysis of a huge sample of DNS queries to three different, heavily used, servers: a proxy resolver, an ISP DNS server, and a root nameserver.

TLS runs over TCP, but DNS has historically used User Datagram Protocol (UDP). The connectionless nature of UDP originally seemed suited to the simplicity of the query/reply nature of DNS. But, there are downsides to this simplicity, one of them being an "amplification attack" in which malicious senders use their victim's IP address to trick DNS servers into sending large replies that overwhelm the victim's network capacity. There's evidence that most denial-of-service attacks utilize DNS amplification. TCP and TLS together thwart a good number of these simple attacks, giving yet another reason to switch away from UDP.

The TLS approach is an interesting contrast to an older proposal for DNS query privacy[5] that works over UDP, but has been slow to catch on. That method seems roughly comparable to this new proposal in terms of performance.

There are two nonobvious implementation changes that the researchers used to argue that TCP is a good choice for a transport protocol. Traditional query/response systems process requests in order, but this blocks the connection until the response is ready. If instead the server delivers replies when they're ready, even if they're out of order, the connection can be kept open without blocking. Because DNS already has provisions for matching responses to queries, this change isn't onerous for clients. The clients can improve things further by sending the queries without waiting for responses; the typical webpage has at least four unique domain names, and all those queries can be sent in one batch. Finally, clients could help servers avoid the expense of tearing down connections and restarting them by advising the server that it would be a good idea to extend the timeout period.

Based on the data available, the researchers feel that two other modifications would help with the rather expensive process of starting up a new TLS connection for DNS. They propose adding a new bit to the extensions vector to indicate that the client wants to use TLS immediately. They also recommend adopting the strategy of letting the server bundle the TLS session state into an opaque object that can be stored by the client while the session is closed. The client could request session resumption

by sending the opaque object back to the server, and both could then restore their cryptographic contexts securely.

How practical is all of this? The paper is based on traffic analyses, not actual implementations, but it suggests that the time to make this a practical reality is at hand. There's even a working group for that: the DNS PRIVate Exchange (dprive).[6]

A Practical Approach to Finding TLS State Machine Errors

The TLS security protocol is widely used for authenticating and protecting connections to webservers. Briefly, it's the lock icon shown next to the address bar in a browser. The protocol has also been the subject of many detailed analyses and the source of many bugs.[7] Problems don't come from the protocol's formal definition, which has been analyzed extensively, but from the implementations that are burdened with legacy functionality, cipher suite variability, and complex error handling. Practically speaking, how can we vet a TLS implementation for correct handling of everything thrown at it?

The problem is especially difficult because there's not just one sequence of messages between a client and server that sets up the cryptographic context for a TLS session. If there were, it would be easy to test for correctness. At each state, a testing system would try sending one of the 10 or so messages associated with a different, out-of-order state transition. But TLS isn't that simple, and there are multiple valid pathways that weave through its state transition graph. Not all implementers coded this correctly, which led to a number of exploitable vulnerabilities.

Can all the errors in TLS implementations be found and corrected? Perhaps not all of them, but Benjamin Beurdouche and his colleagues[8] have an innovative testing methodology that automatically detects problems in real-life implementations of TLS. They probe at an implementation (either a client or a server) by sending messages out of order and looking for responses that fail to "alert" the bad message. This was a surprisingly fruitful investigation. For example, they found that one implementation incorrectly allowed some cryptographic setup messages to be skipped. In one case, this omission disabled subsequent data encryption.

One of the discoveries was an exploitable downgrade attack that has been named "FREAK." The problem comes about because some client implementations incorrectly accept a key exchange message, even though they already have the server's RSA key from its certificate. The key exchange message can have a very weak RSA key (from the days when US export controls required weakness in exported cryptographic products). As a result, a man-in-the-middle (MITM) can factor the key and read the traffic. In fact, for many implementations, the MITM can read the traffic for several days, because the key is changed infrequently.

A second part of the research produced a verified implementation of the complete TLS state machine. Their implementation requires annotations on critical memory areas to assure that they're well-defined and not overlapping. This verified implementation could be used within another implementation as a check that each protocol message left the implementation in a valid state.

As practical as this seems to be, as a way of locking down the TLS state machine, the authors note that the whole protocol, including all the cryptography, is unlikely to be verified anytime soon. Still, we can only applaud any method that finds bugs for the good guys to fix before the bad guys find them.

Capabilities Made (Almost) Practical

Finally, wouldn't it be nice if the computer hardware protected processes from all unauthorized data accesses, rendered buffer overflow bugs innocuous, and generally kept us all safe? Of course it would, and that idea is behind the venerable capability machine which has long been the Holy Grail of the secure processing community.

Capability machines use special hardware to mediate interactions between different "domains" (roughly speaking, processes and/or libraries). A *capability architecture* can securely implement the principle of "least privilege" for trusted computing bases. A capability is an immutable system object that we can use for resource access. For example, we could control read access to a section of memory from another process through a capability rather than a bare memory pointer or kernel-mediated interprocess communication. Capabilities are also useful for controlling access to resources such as files or network connections.

The problem that has beset capability machines from their outset is that they're slow. Compared to traditional processors of similar instruction sets, they seem like sack race contestants trying to run Olympic sprints. That slowness comes from the number of independent address space descriptors needed to represent a capability-based process. Each one requires a memory-page mapping. The translation from address to page map index is done via the fast associative memory cache structure called the *translation lookaside buffer* or TLB (which is generally as well understood as a spleen). If there are more active entries than can fit in the TLB, memory lookups go through a much slower lookup mechanism that makes a modern computer seem to be running in low gear.

Undissuaded by this history, a large research team has embarked on the Capability Hardware-Enhanced RISC

Instructions (CHERI) project to "do capabilities right" on their Business Environment Risk Intelligence (BERI) processor, a RISC field-programmable gate array (FPGA) softcore processor (open sourced). Their recent work[9] builds on a capability architecture, but sidesteps some of its associative memory limitations by way of a hybrid approach that selectively integrates a C compiler with software capabilities on a FreeBSD (BSD stands for Berkeley Software Distribution) operating system. If that sounds familiar, it's because the capabilities of FreeBSD were the subject of work published in 2010.[10] That paper investigated the "capabilitization" of common utilities like `tcpdump` and `zlib`.

The work described at the symposium is an interesting architecture that lets a developer choose security/performance tradeoffs in a capability architecture. At one end of the spectrum, everything could be a capability: all library accesses, all memory pointers, all interprocess communication. That would be prohibitively expensive, as noted in the past. That's why CHERI supports intermediate solutions that can set the protection boundaries in several different ways. Capabilities can be used to wrap an untrusted library operating within an otherwise unmodified application, for example. Or, a capability-based process can access "compartmentalized" legacy C code. The compartment limits the damage from buffer overflows or improper addressing to the resources of the software and data within the compartment's address space.

The performance results are greatly favorable to the capabilities when used correctly, as opposed to sandboxing. Interprocess memory copies, for example, are much speedier with a capability than with sandboxed processes. The `tcpdump` utility is an interesting case study, because it has known bugs when handling some kinds of malformed packets. CHERI is flexible enough to support isolation based on packet types or addresses, packet processing time, or other criteria. An error in a compartment is limited to damage within its own scope. These techniques mitigated all but two known vulnerabilities in the utility.

The CHERI work isn't ready for immediate use, and their FPGA processor isn't going to be fabricated for use in mobile devices anytime soon. Yet the work does show that stronger protections for real-world legacy code could be a reality in a future world that values secure processing. As a practical matter, security-enhanced processors still seem a distant hope, unless there's some overwhelmingly lucrative use for them. This seems a bit odd, because the estimated losses from cybercrime are astounding.

There's no crystal ball that tells us when research will turn into practice. Although computer security research is very much an applied science (I'm reminded of Edsger Dykstra's cutting remark that computer science is no more a science than surgery is "knife science"), and even though most researchers would be delighted to have their work used commercially, the pathway from the research lab to the sales-in-the-billions product is unpredictable. Economics, physics, and psychology must all align perfectly. The four examples presented here are novel approaches expanding computer security, and each of them represent ongoing work that could be in your hands within a few years. They're only a small sample of what comes out of academic and industrial research labs every year, so keep your eye out for the gems that might be the next big influence on security.

References

1. K. Thomas et al., "Ad Injection at Scale: Assessing Deceptive Advertisement Modifications," *Proc. IEEE Symp. Security and Privacy*, 2015, pp. 151–167.
2. S. Rosenblatt, "Lenovo's Superfish Security Snafu Blows Up in Its Face," *CNET*, 20 Feb. 2015; www.cnet.com/news//superfish-torments-lenovo-owners-with-more-than-adware.
3. Y. Zhou and D. Evans, "Understanding and Monitoring Embedded Web Scripts," *Proc. Symp. IEEE Security and Privacy*, 2015, pp. 850–865.
4. L. Zhu et al., "Connection-Oriented DNS to Improve Privacy and Security," *Proc. IEEE Symp. Security and Privacy*, 2015, pp. 171–186.
5. M. Dempsky, *DNSCurve: Link-Level Security for the Domain Name System*, IETF draft, 26 Feb. 2010; http://tools.ietf.org/html/draft-dempsky-dnscurve-01.
6. DNS PRIVate Exchange (dprive), *Charter for Working Group*, 2015; https://datatracker.ietf.org/wg/dprive/charter.
7. C. Meyer and J. Schwenk, *Lessons Learned from Previous SSL/TLS Attacks – A Brief Chronology of Attacks and Weaknesses*, Int'l Assoc. for Cryptologic Research (IACR) eprint archive, 2013; http://eprint.iacr.org/2013/049.
8. B. Beurdouche et al., "A Messy State of the Union: Taming the Composite State Machines of TLS," *Proc. Symp. IEEE Security and Privacy*, 2015, pp. 535–552.
9. R.N.M. Watson et al., "CHERI: A Hybrid Capability-System Architecture for Scalable Software Compartmentalization," *Proc. IEEE Security and Privacy Symp.*, 2015, pp. 20–37.
10. R.N.M. Watson et al., "Capsicum: Practical Capabilities for Unix," *Proc. 19th Usenix Security Symp.*, 2010; www.usenix.org/legacy/event/sec10/tech/full_papers/Watson.pdf.

Hilarie Orman is a security consultant and president of Purple Streak. Her research interests include applied cryptography, secure operating systems, malware identification, security through semantic computing, and personal data mining. Orman has a BS in mathematics from the Massachusetts Institute of Technology. She's a former chair of the IEEE Computer Society's Technical Committee on Security and Privacy. Contact her at hilarie@purplestreak.com.

Login to mycs.computer.org

READ YOUR FAVORITE PUBLICATIONS YOUR WAY

Now, your IEEE Computer Society technical publications aren't just the most informative and state-of-the-art magazines and journals in the field — they're also the most exciting, interactive, and customizable to your reading preferences.

The new myCS format for all IEEE Computer Society digital publications is:

- **Mobile friendly.** Looks great on any device — mobile, tablet, laptop, or desktop.
- **Customizable.** Whatever your e-reader lets you do, you can do on myCS. Change the page color, text size, or layout; even use annotations or an integrated dictionary — it's up to you!
- **Adaptive.** Designed specifically for digital delivery and readability.
- **Personal.** Save all your issues and search or retrieve them quickly on your personal myCS site.

You've Got to See It!

To really appreciate the vast difference in reading enjoyment that myCS represents, you need to see a video demonstration and then try out the interactivity for yourself. Just go to **www.computer.org/mycs-info**.

IEEE computer society

Editor: Jonathan Margulies, jonathan@qmulos.com

Colluding Apps:

Tomorrow's Mobile Malware Threat

Atif M. Memon | University of Maryland, College Park
Ali Anwar | Montgomery Blair High School

Mobile devices (tablets, smartphones, watches, and wearable gadgets) carry a wealth of personal and professional data that software apps can read, access, and modify. Unfortunately, some apps are malicious, stealing users' data and transmitting it without their knowledge. Given that an estimated 7.22 billion mobile devices are in use—a number rivaling the human population[1]—and that mobile platforms are increasingly reporting malware,[2,3] mobile malware might put our privacy at unprecedented risk.

App Isolation and Interaction

Today's mobile OSs are designed with a focus on security. Android, the most popular mobile OS, isolates each app in an *application sandbox* by leveraging the security features of its underlying Linux kernel. Each app runs in its own memory space, has access to a permission-protected file system, and has protected CPU cycles. Unless the user explicitly bypasses it, this sandbox design protects apps from interfering or interacting with one another and other vital system components. For example, a banking app can't access files from a messaging app and vice versa.

Although it successfully isolates apps from one another, the sandbox design doesn't completely preclude malware exploitation. For example, an app might intentionally or unintentionally leak the device's GPS coordinates by encoding them in the URL of an HTTP request. Android provides additional mechanisms to protect against such leaks. Each protected feature of the device (such as GPS or the network) requires explicit access permission. Hence, for an app to leak GPS location information over the network, it must have simultaneous permission to ACCESS_FINE_LOCATION for GPS and access to INTERNET for the network. Only users can grant these permissions when they install the app. However, because most users ignore permission warnings at app install time,[4] they might end up installing overprivileged apps—those requesting more permissions than they need to do their job[5]—some of which might be malicious.

It's not practical to completely isolate apps from one another and the system. Numerous use cases require that apps be allowed (and even encouraged) to interact; for example, a messaging app might need to interact with the device contacts app and the camera app so users can capture and send pictures to their contacts. Moreover, apps might invoke parts of other apps to enhance the user experience. For example, an app that wants to show a map as part of its user interface doesn't need to write map-viewing code from scratch. Instead, it can

2376-113X/16/$33.00 © 2016 IEEE Published by the IEEE Computer Society

Covert Communication Channels

Consider two apps: *FFitt* and *IIMsgg*. FFitt collects and maintains fitness data locally on the user's mobile device. It has permission to communicate unshared local storage, device location, and system settings via Bluetooth to a fitness device. It has no Internet access. IIMsgg sends and receives text messages via an Internet-based messaging service. It has permission to access the Internet, contacts, and system settings. Security analysis tools would deem both apps safe because there's no way to leak sensitive fitness and location data.

Unbeknownst to these analysis tools is that there's malicious code (a snippet is shown in Figure A) embedded in the apps by developers or a hacked integrated development environment[1] that allows them to communicate via an unconventional channel: the device's screen-off time-out. In Android, this is the time in milliseconds before the device goes to sleep or begins to dream after a period of inactivity. Apps are allowed to read and modify this value. Using this mechanism, the FFitt app covertly sends fitness and location data to the IIMsgg app, which then leaks this information to a contact via a message. Figure A implements an oversimplified but illustrative protocol that allows FFitt to transmit its sensitive data as a set of numbers (`message` in the code). FFitt encodes each number as a time-out value and sets it using `Settings.System.putInt` (full code not shown due to space); IIMsgg reads the value (`Settings.System.getInt`) and resets it to indicate successful receipt. FFitt then "sends" the next number. This process continues until all the numbers have been transmitted. This example illustrates how two apps' shared resource (screen time-out) might be used as a covert communication channel to effectively complete a path from a source of sensitive data (location, fitness data) to an external sink. None of today's malware tools would detect this collusion.[2]

References

1. "Novel Malware XCodeGhost Modifies XCode, Infects Apple iOS Apps and Hits App Store," Palo Alto Networks, 17 Sept. 2015; http://researchcenter.paloaltonetworks.com/2015/09/novel-malware-xcodeghost-modifies-xcode-infects-apple-ios-apps-and-hits-app-store.
2. D.J.J. Sufatrio et al., "Securing Android: A Survey, Taxonomy, and Challenges," *ACM Computing Surveys*, vol. 47, no. 4, 2015, pp. 58:1–58:45.

invoke the default map app's map screen (similar to a subroutine); users can then view the map, close it, and return control back to the calling app. In another example, users might want a certain app to be invoked for a system event such as a text's arrival.

To cater to such use cases, Android allows apps to interact with one another and the system, pass data, return results, and share resources; apps need to explicitly allow for such interaction via permissions. However, allowing interaction such as communication via messages (implemented as Intent objects in Android) opens the door to malware. For example, the DroidDreamLight malware used receipt of the android.intent.action.PHONE_STATE Intent as its trigger—such as when users receive a phone call. Once triggered, this malware executes its own code.[6]

Detecting and Removing Mobile Malware

The Internet abounds with advice and best practices for avoiding mobile malware. Yet malware continues to proliferate, indicating that prevention alone isn't sufficient. Several techniques can detect and remove mobile malware. First, *basic static techniques* check the app's attributes such as file name, checksums or hashes, file type, and file sizes. Any discrepancies are flagged as potential malware. Second, *static code search techniques*, such as those used in virus scanners, search for syntactic signatures in the app's code. Using a database of code sequence patterns that are

Figure A. A snippet of malicious code embedded in the FFitt and IIMsgg apps that lets them communicate via an unconventional channel: the device's screen-off time-out. FFitt encodes sensitive fitness and location data as time-out values, which IIMsgg then sends to a contact via a message.

considered malicious, the technique identifies a program as malware if part of its code matches a pattern in the database. Third, *static code analysis techniques* perform flow analysis of the app's code to check whether there's a control flow path that allows the app to access and leak sensitive information to an external entity.

These static techniques are severely limited because today's mobile apps increasingly use dynamic runtime mechanisms such as virtual function calls, dynamic class loading, reflection, multithreading, and event handler callbacks. Such mechanisms necessitate a fourth technique: *dynamic analysis*. This technique manually or automatically runs the app and observes its runtime behavior; for example, it monitors system processes, file-system and registry changes, and network activity. Dynamic analysis is typically done in a safe (most likely emulated) environment in which the analyst can run the malware and observe its behavior without interference from other apps. However, this technique is incomplete because it's generally impossible to run a software program on all its possible inputs. Moreover, because dynamic analysis involves actually executing the app, the app might fool the analysis by "turning off" malicious behavior in certain configurations (such as for particular platforms, locations, time, and devices). For example, the Dendroid malware used emulation-detection code to successfully evade Google Bouncer, an automated app-vetting tool.[7]

App Collusion

Even as the security community is starting to understand and detect individual malicious apps, a new threat is emerging: *colluding apps*.[8] In collusion attacks, a malicious operation is broken into smaller parts and distributed across multiple apps. These apps communicate (or wait for a signal) to play their small, individually undetectable roles in the operation. Each app avoids suspicion by requesting the minimum permissions needed for its role. For example, when two apps collude, the first app might read sensitive data and transmit it to the second app, which transmits it to the outside world. Analyzed individually, the apps would be considered benign because there's no direct path from sensitive data to its transmission. In an effort to detect colluding apps, recent research has extended flow analysis to include app message-passing channels.[9] These messages can be monitored at runtime to identify app pairs that exchange messages and, hence, possibly collude.[10]

In the "Covert Communication Channels" sidebar, the example of collusion between the FFitt and IIMsgg apps demonstrates the serious and complex nature of collusion. In 1973, Butler Lampson identified a general form of this "confinement problem,"[11] although its application to today's mobile apps gives it new light. Malware detection tools considering each app in isolation have no hope of detecting collusion because they aren't designed to analyze sets of apps simultaneously. Even if new tools could target collusion, they'd be limited in at least two ways. First, they wouldn't know which apps to analyze together: there are millions of apps in the marketplace, and any two (or more) might be malicious and colluding. To analyze all possible app pairs, the tool would need to analyze N^2 pairs, where N is the number of apps in the marketplace. Analyzing all possible triples—to detect sets of three colluding apps—would require N^3 runs. Thus, the cost of analysis grows exponentially with the number of concurrently analyzed apps. Second, the analysis wouldn't know which communication channels to intercept.

Apps share dozens of resources, each with multiple attribute values that apps are allowed to read or modify (including network status, sound volume level, device orientation, Bluetooth status, USB connection, and altitude). Any of these resources might be used—individually or together—as covert communication channels. In principle, a successful analysis tool must monitor all possible communication channels, which means monitoring every possible code path to every read/write of a shared resource's attribute. This is a practically impossible task.

Potential Security Improvements

The problem of colluding malware hits at the very core of today's mobile OS security model: individually restricting apps (for example, via permissions or sandboxing) is sufficient for their safe composition on a single device. This model is inadequate in light of colluding malware. Thus, we must revise and enhance the model, which will involve an expensive, major rewrite of the security components of today's mobile OSs. Detecting colluding apps remains an open research problem, the solution to which eludes both practitioners and researchers. We believe that any solution must address two fundamental challenges: which covert channels to examine and which sets of apps to analyze together for collusion.

To identify covert channels, we propose the following steps:

- Exhaustively list all possible shared resources and their attributes that apps can access and modify (originally called "shared-resource matrix methodology"[12]).
- Examine the code of all apps in today's mobile marketplace to determine the shared resources and attributes being accessed in practice.

To identify which sets of apps to analyze together for collusion, we propose the following approaches:

- Examine app advertisements that ask users to download other apps.
- Mine social media postings and Internet sites that ask users to download apps. For example, 3 million Minecraft fans—looking for cheat sheets and playing tips—were tricked into downloading more than 30 fake apps.[13]

There's no way to tell how many covert-channel colluding apps are in the mobile marketplace, already stealing our information. Many users are unaware that their devices have even been compromised. Indeed, if rogue nations use malware to spy on other countries' government agents, businesses, and diplomats to gain strategic advantage, then the results of exploits might not translate to a traceable outcome (such as a credit card charge)—meaning that the malware can go undetected for years.

Given that shared resource attributes are easily repurposed as covert communication channels, we believe that we're on the edge of a massive influx of apps using such channels to go undetected for long periods. It's not difficult to imagine the developers of today's individual malicious apps splitting their malware code across multiple and seemingly benign apps, using a covert channel to communicate between the apps, and successfully performing malicious operations. If

their exploits are discovered, they can quickly move the malware to other apps, change the covert communication to an alternative shared resource, and repenetrate the marketplace. This practice could continue indefinitely, until we develop better solutions to the problem of malware avoidance and detection. ■

References

1. "There Are Officially More Mobile Devices than People in the World," *Independent*, 7 Oct. 2014; www.independent.co.uk/life-style/gadgets-and-tech/news/there-are-officially-more-mobile-devices-than-people-in-the-world-9780518.html.
2. "Protect Your Android Device from Malware," CNET, 25 June 2014; www.cnet.com/how-to/protect-your-android-device-from-malware.
3. "McAfee Labs Threats Report May 2015," McAfee, May 2015; www.mcafee.com/us/resources/reports/rp-quarterly-threat-q1-2015.pdf.
4. A. Porter Felt et al., "Android Permissions: User Attention, Comprehension, and Behavior," *Proc. 8th Symp. Usable Privacy and Security* (SOUPS 12), 2012, pp. 3:1–3:14.
5. X. Wei et al., "Permission Evolution in the Android Ecosystem," *Proc. 28th Ann. Computer Security Applications Conf.* (ACSAC 12), 2012, pp. 31–40.
6. M. Balanza et al., "DroidDream-Light Lurks behind Legitimate Android Apps," *Proc. 6th Int'l Conf. Malicious and Unwanted Software* (MALWARE 11), 2011, pp. 73–78.
7. "Dendroid Spying RAT Malware Found on Google Play," Help Net Security, 3 July 2014; www.net-security.org/malware_news.php?id=2726.
8. C. Marforio et al., "Analysis of the Communication between Colluding Applications on Modern Smartphones," *Proc. 28th Annual Computer Security Applications Conf.* (ACSAC 12), 2012, pp. 51–60.
9. D. Sbîrlea et al., "Automatic Detection of Inter-application Permission Leaks in Android Applications," *IBM J. Research and Development*, vol. 57, no. 6, 2013, pp. 2:10– 2:10.
10. K.O. Elish, D. Yao, and B.G, Ryder, "On the Need of Precise Inter-app ICC Classification for Detecting Android Malware Collusions," *Proc. IEEE Mobile Security Technologies* (MoST)/*IEEE Symp. Security and Privacy* (SP), 2015.
11. B.W. Lampson, "A Note on the Confinement Problem," *Comm. ACM*, vol. 16, no. 10, 1973, pp. 613–615.
12. R.A. Kemmerer, "Shared Resource Matrix Methodology: An Approach to Identifying Storage and Timing Channels," *ACM Trans. Computer Systems*, vol. 1, no. 3, 1983, pp. 256–277.
13. "Minecraft Cheats Scareware Apps Affect 600,000 Users," WCCF Tech, May 2015; http://wccftech.com/minecraft-cheats-scareware-apps-affect-600000-users.

Atif M. Memon is a professor in the Department of Computer Science and the Institute for Advanced Computer Studies at the University of Maryland, College Park. Contact him at atif@cs.umd.edu.

Ali Anwar is a student at Montgomery Blair High School. Contact him at alia7477@gmail.com.

This article originally appeared in
IEEE Security & Privacy, *vol. 13, no. 6, 2015.*

RICHARD E. MERWIN SCHOLARSHIP

IEEE COMPUTER SOCIETY RICHARD E. MERWIN STUDENT LEADERSHIP SCHOLARSHIP

$40,000 AVAILABLE FOR IEEE COMPUTER SOCIETY MEMBERS

Richard E. Merwin Scholarships are awarded to recognize and reward active student volunteer leaders who show promise in their academic and professional efforts.

- Scholarships from $1,000
- Recipients are recognized as Computer Society Ambassadors and receive mentoring opportunities
- Available to graduate and undergraduate students in their final two years
- Students must be enrolled in a program in electrical or computer engineering, computer science, information technology, or a well defined computer-related field
- IEEE Computer Society membership required at time of application

IEEE computer society

www.computer.org/merwin

▶ DEADLINE: 30 APRIL 2016

SOME OF OUR PAST RICHARD E. MERWIN SCHOLARSHIP WINNERS …

Ambika Shivana Jagmohansingh University of the West Indies (Jamaica) | **Amente Bekele** Carleton University (Canada) | **Atefeh Khosravi** University of Melbourne (Australia) | **Christen M. Corrado** Rowan University (USA) | **Irene Mathew Susan** College of Engineering, Chengannur (India) | **Josip Balen** J.J. Strossmayer University of Osijek (Croatia) | **Marios Bikos** University of Patras (Greece)

Cryptography Is Harder than It Looks

Bruce Schneier
Resilient Systems

Writing a magazine column is always an exercise in time travel. I'm writing these words in early December. You're reading them in February. This means anything that's news as I write this will be old hat in two months, and anything that's news to you hasn't happened yet as I'm writing.

This past November, a group of researchers found some serious vulnerabilities in an encryption protocol that I, and probably most of you, use regularly. The group alerted the vendor, who is currently working to update the protocol and patch the vulnerabilities. The news will probably go public in the middle of February, unless the vendor successfully pleads for more time to finish their security patch. Until then, I've agreed not to talk about the specifics.

I'm writing about this now because these vulnerabilities illustrate two very important truisms about encryption and the current debate about adding back doors to security products:

1. Cryptography is harder than it looks.
2. Complexity is the worst enemy of security.

These aren't new truisms. I wrote about the first in 1997 and the second in 1999. I've talked about them both in *Secrets and Lies* (2000) and *Practical Cryptography* (2003). They've been proven true again and again, as security vulnerabilities are discovered in cryptographic system after cryptographic system. They're both still true today.

Cryptography is harder than it looks, primarily because it looks like math. Both algorithms and protocols can be precisely defined and analyzed. This isn't easy, and there's a lot of insecure crypto out there, but we cryptographers have gotten pretty good at getting this part right. However, math has no agency; it can't actually secure anything. For cryptography to work, it needs to be written in software, embedded in a larger software system, managed by an operating system, run on hardware, connected to a network, and configured and operated by users. Each of these steps brings with it difficulties and vulnerabilities.

Although cryptography gives an inherent mathematical advantage to the defender, computer and network security are much more balanced. Again and again, we find vulnerabilities not in the underlying mathematics, but in all this other stuff. It's far easier for an attacker to bypass cryptography by exploiting a vulnerability in the system than it is to break the mathematics. This has been true for decades, and it's one of the lessons that Edward Snowden reiterated.

The second truism is that complexity is still the worst enemy of security. The more complex a system is, the more lines of code, interactions with other systems, configuration options, and vulnerabilities there are. Implementing cryptography involves getting everything right, and the more complexity there is, the more there is to get wrong.

Vulnerabilities come from options within a system, interactions between systems, interfaces between users and systems—everywhere. If good security comes from careful analysis of specifications, source code, and systems, then a complex system is more difficult and more expensive to analyze. We simply don't know how to securely engineer anything but the simplest of systems.

I often refer to this quote, sometimes attributed to Albert Einstein and sometimes to Yogi Berra: "In theory, theory and practice are the same. In practice, they are not."

These truisms are directly relevant to the current debate about adding back doors to encryption products. Many governments—from China to the US and the UK—want the ability to decrypt data and communications without users' knowledge or consent. Almost all computer security experts have two arguments against this idea: first, adding this back door makes the system vulnerable

2376-113X/16/$33.00 © 2016 IEEE Published by the IEEE Computer Society

This article originally appeared in
IEEE Security & Privacy, *vol. 14, no. 1, 2016.*

to all attackers and doesn't just provide surreptitious access for the "good guys," and second, creating this sort of access greatly increases the underlying system's complexity, exponentially increasing the possibility of getting the security wrong and introducing new vulnerabilities.

Going back to the new vulnerability that you'll learn about in mid-February, the lead researcher wrote to me: "If anyone tells you that [the vendor] can just 'tweak' the system a little bit to add key escrow or to man-in-the-middle specific users, they need to spend a few days watching the authentication dance between [the client device/software] and the umpteen servers it talks to just to log into the network. I'm frankly amazed that any of it works at all, and you couldn't pay me enough to tamper with any of it." This is an important piece of wisdom.

The designers of this system aren't novices. They're an experienced team with some of the best security engineers in the field. If these guys can't get the security right, just imagine how much worse it is for smaller companies without this team's level of expertise and resources. Now imagine how much worse it would be if you added a government-mandated back door. There are more opportunities to get security wrong, and more engineering teams without the time and expertise necessary to get it right. It's not a recipe for security.

Unlike what much of today's political rhetoric says, strong cryptography is essential for our information security. It's how we protect our information and our networks from hackers, criminals, foreign governments, and terrorists. Security vulnerabilities, whether deliberate backdoor access mechanisms or accidental flaws, make us all less secure. Getting security right is harder than it looks, and our best chance is to make the cryptography as simple and public as possible. ■

Bruce Schneier is the CTO of Resilient Systems. His new book is *Data and Goliath: The Hidden Battles to Collect Your Data and Control Your World.* Contact him via www.schneier.com.

Editor in Chief: **Diomidis Spinellis**
Athens University of Economics
and Business, dds@computer.org

Developer, Debug Thyself

Diomidis Spinellis

IT WAS A NEAR MISS. The press portrayed Volkswagen as the corporate villain that manipulated its diesel engines' performance during emissions tests. Yet, we software professionals know that a developer ultimately implemented the code that detected when the tests were being conducted and changed the engine's behavior accordingly. On the next scandal our profession won't escape unscathed. As professionals, we're already on the hook for developing software that regularly leaks embarrassing amounts of personal data, allows the flourishing of illegal botnets that control millions of PCs, and has provided backdoors to government agencies. Clearly, we must deal more effectively with our professional responsibility.

Continuing business as usual isn't an option. Software increasingly affects our personal lives, the economy, society, and our planet. Software engineers are daily called to participate in tricky decisions at the crossroads between ethics and more narrowly defined corporate interests. For example, consider a smartphone. Should it click when taking pictures, thereby alerting others that they're being photographed? (Mine doesn't when in silent mode.) What should it do when recording video? Should it raise the headphone audio to harmful levels? (Mine issues a warning but lets me override it.) Should it offer the option to record phone calls? How should it handle its owner's personal data? Should the engineers' allegiance be to the phone's owner or the company's shareholders? Should the phone offer a backdoor to law enforcement authorities? Should that backdoor's existence be kept secret?

Software increasingly affects our personal lives, the economy, society, and the planet.

The questions we face are difficult and will become even more so. If you think it's easy to answer the questions I outlined because a smartphone's legal owner should have some clear rights, reflect on the case of the customers using "free" social-networking applications. Also keep in mind that these questions might be more difficult to answer when a product or service is made available in totalitarian countries that regularly violate human rights. And the list of areas in which software plays a crucial role keeps growing. Apart from the well-known areas of transport, medical devices, and nuclear energy, consider that software in self-driving cars, wearables, entertainment, and the smart grid easily raises even thornier questions.

Any response dealing with software developers' professional responsibility should take into account the software's complete life cycle.

IEEE Software Mission Statement

To be the best source of reliable, useful, peer-reviewed information for leading software practitioners—the developers and managers who want to keep up with rapid technology change.

Going back to the Volkswagen case, there were probably people who specified the requirement for the engine test manipulation, designed it, modeled it, constructed it, tested it, put it under configuration management, and passed it through quality assurance. Managers, development leads, and programmers allowed the mischief to happen.

> Organizations that are found to flout the rules should be forced to release their software as open source.

The Problems of Regulation

If we don't act decisively and effectively through our professional organizations, such as the IEEE Computer Society and the ACM, governments will step in with thickets of regulation. This outcome will be catastrophic to our field, ending a half century of magnificent innovation and growth that have provided countless benefits to mankind.

Excessive government regulation of the rapidly evolving field of software development will be destructive because the precise and rigid oversight of software products and processes will stifle innovation. This is already happening in many other industries, including the control of car emissions, in which targets are rigidly set through unrealistic test scenarios. Instead, some people have argued that environmental outcomes could be improved at a lower cost by addressing the problem systemically through the use of GPS and information technology to differentiate the engine parameters of cars driving in a congested city center from those of cars driving on a desert highway. These are optimizations that large and small IT companies, from Amazon to Zimbra, make daily as part of their business.

Under a tight regulation regime, we can easily imagine governments gumming up the global software industry by micromanaging product specifications and auditable deliverables between development processes. In addition, in some countries, tight regulation will breed corruption, whereas other countries will use regulation to erect new trade barriers. All the while, these countries will lose their intellectual capital as software innovation moves to those with more enlightened regulation regimes.

Regulation will also likely be ineffectual. The huge amount of software available today, coupled with the practice of end-user programming and the fact that software is woven in thin air and thus difficult to control, will leave any regulations that can be applied in practice too vague and full of loopholes.

Governments are trying to sidestep these problems by vertically regulating specific industries, as is the case for medical devices. However, this is becoming more and more difficult as software is increasingly interlinked into complex large systems of systems with fluid, permeable boundaries. Consider an implanted pacemaker that a patient can monitor through a phone app and that a physician can fine-tune remotely. What will be regulated as medical software apart from the pacemaker's code? The tuning software? The firmware on the physician's broadband router? The patient's phone app? The phone's OS? Other apps on the phone? All these entities could be maliciously used to attack the pacemaker. And if you think that strict regulation is warranted because it's better to be safe than sorry, take into account the respected voices who now argue that overregulation of drug trials often hurts patients by delaying the use of life-saving medicines.

Better Alternatives

Instead of increased government meddling, a more effective and efficient approach would be a well-functioning regime of self-regulation. Professional bodies would set broad standards of professional conduct and practice related to software development, and developers would be bound to follow them and would be responsible for their actions. Governments would oversee the regime's monitoring and enforcement.

Getting such a system to work won't be trivial. Current regimes of accreditation, certification, and licensing[1] must be adjusted to cover all areas where intervention is needed. Standardization must be extended to cover any important gaps. Software professionals' education and training must be updated to deal with these changes. Codes of ethics and professional conduct must be better communicated and enforced through actions that will have teeth. Sadly, up to now, the reaction of professional

societies regarding the engineers responsible for the Volkswagen case has been missing in action.

Transparency can also help. In code developed as open source, developers are less likely to commit shenanigans, especially if they know that from day one their contributions will be openly available, as code commits signed with their name. Furthermore, open source software exposes what's technically possible and what's not, providing the government and society with the reliable information needed to make policy decisions regarding the software's desired attributes. In contrast, proprietary software obscures true technical affordances, providing many people with a false sense of security while only marginally inconveniencing criminals. Examples of this phenomenon include ostensibly secure systems of lawful interception and digital rights management.

Transparency can also help motivate compliance: organizations that are found to flout the rules should be forced to release their software as open source. This would be a powerful deterrent, while providing the research community and professional societies with material we can use to improve our understanding of compliance monitoring and enforcement.

The risks of misbehaving software have been with us for decades but are now becoming too ubiquitous to casually brush under the carpet. We must act now; otherwise, the next software scandal might take down software development as we know it.

Reference

1. P. Kruchten, "Licensing Software Engineers?," *IEEE Software*, vol. 25, no. 6, 2008, pp. 35–37.

This article originally appeared in IEEE Software, *vol. 33, no. 1, 2016.*

Aisling Kelliher
Virginia Tech

Small Data, Big Impact

Dane Webster and Ivica Ico Bukvic
Virginia Tech

In a world of big data, sometimes having the right amount of small data at the right time can also pack a powerful punch. Tasked with creating a straightforward network visualization for a growing institution, we began an exploration of the organization's activities that ultimately led to a radical refocusing and restructuring of the institution itself. Our small interdisciplinary team, combining skills in data visualization, music composition, and computational programing, produced the *Orb*—a data visualization project leveraging perceived affordances of 3D data representation to support human understanding of institutional networks. Here, we present our multistage design process and lessons learned from this ongoing project.

ICAT: An Evolving Institution

The Institute for Creativity, Arts, and Technology (ICAT) is one of seven institutes at Virginia Tech charged with tackling large-scale research problems through interdisciplinary collaboration. Although formally launched in October 2011, the evolution of ICAT has a storied and ongoing development trajectory, lending itself well to an investigation of organizational dynamics emerging over time.

The current ICAT mission statement declares that it "will forge a pathway between transdisciplinary research and artistic output, scientific and commercial discovery, and educational innovation" by fostering the creative process in creating "new possibilities for exploration and expression" (see www.icat.vt.edu). The path to crafting this mission began more than 10 years ago with a cluster-hire initiative at Virginia Tech across four departments: Art, Music, Computer Science, and Cinema. Four faculty members were hired within these departments with the express goal of developing cross-disciplinary, collaborative projects that helped breach institutional boundaries and silos.

As projects and the number of collaborating partners grew, the idea of developing an encompassing institution to conduct this type of work at scale also emerged. The initial concept was to structure the institute's activities around focused research studios, whereby the use of the term "studio" as opposed to "lab" served to emphasize the creative components and outcomes of the work carried out. As these studies flourished over time, Ben Knapp, the director of ICAT, asked us if we would be interested in creating a quick visualization graphic of the emerging institutional structure. From this initial informal request, we expanded the concept to encompass dynamic and interactive elements, leading to the design and development of the Orb.

The Orb

The Orb is a data visualization tool (see Figure 1) created using the open source development environment Processing (https://processing.org). The project has come through two major development cycles: building the core visual structure, and extending its functionality by integrating into the structure an aural component and applying external sensor technologies to support gestural interaction.

The Orb visualizes the "connectedness" of the various research initiatives associated within ICAT, whereby the connectedness is calculated based on an analysis of the keywords and descriptions associated with each research project. The viewer is presented initially with an interactive representation of a 3D sphere with individual nodes along its surface. The viewer can then select the individual nodes to view written descriptions and movie files showcasing the work. Weighted values are assigned to each project based on their relevancy to any other project, so when any one project is selected, arching lines that meet a certain threshold value are drawn from that selected project to those that are most similar to it.

Keep the Design, Change the Organization

Illustrating the interconnectedness of the institution's various research, outreach, and creative

 Published by the IEEE Computer Society

projects was central to our design methodology for the Orb. The use of a 3D sphere as a hub for the visual interface—a visual representation that adds a certain level of complexity to the interface design—was intentional from the outset. Using the 3D form provided a dynamic multidirectionality that is symbolic of the transdisciplinary research space.

Our decision to organize the visualization generally around projects (as opposed to the foundational studio model) created several design challenges and, indeed, disagreements. During an early critique session, some stakeholders pushed for the studio projects to be organized into defined areas within the Orb. For example, projects associated with one particular studio might be located at the top of the sphere, with a different studio having its projects falling between its own defined latitudes along the Orb. These stakeholders suggested flipping the design methodology, setting the defined boundaries of the studios at the center of the visualization, but this approach did not fit our designed structure of the visualization tool.

In working toward a design solution, we had inadvertently uncovered a more pressing issue about ICAT's organizational structure—whether projects or an organizational structure (the studios) should define the next steps in the institute's evolution.

With the benefit of hindsight, it could be argued that a valid measure of success for such a transdisciplinary research institute should be the frequency of projects that are difficult to fit within clearly defined boundaries of particular institutionalized research areas—or studios in the case of ICAT. But when setting out to create a new organization of any type, it is only natural to want to define categories and assign various members of your team the responsibility of overseeing these particular areas. As we iterated through various versions of the Orb, it became clear to the various stakeholders that the existing organizational structure had outlived its usefulness, and a more diffuse structure was required. The notion of a studio lead was replaced with that of a senior fellow, who, rather than being responsible for a defined area, is pushed to develop projects and make connections to any and all relevant collaborators.

Deployment in a Public Space

A second phase of our project involves retrofitting the existing visualization with an interactive kiosk-like interface placed in a public transitional space. Sponsored by the Virginia Tech National Capital Region (VTNCR), this ongoing iteration aims to provide a series of visual catalysts to encourage passerby participation using different interaction interfaces (such as Kinect or Leap Motion), and engage audiences in the exploration of (in this case) cross-referenced information on VTNCR's research and outreach efforts.

Figure 1. Photo of the Orb interface. Here, the selected project is the Center for the Arts, with arching blue lines indicating other closely related projects.

Creating the Interface

In creating this robust, remotely maintainable audio-visual data experience, on the front end, we must further

- broaden the visual vocabulary to reflect new possible interaction states (such as a standby mode, which resets the Orb into a default state, waiting for a new user interaction);
- streamline the user experience to better match the kiosk format and gesture-based human-computer interaction; and
- inject an aural framework into the application to support the multimedia project materials (such as video footage).

In addition, to enhance the overall experience and ease the cognitive load,[1] we plan to associate specific actions with a vocabulary of *earcons*—an audio counterpart to an icon.[2] We will pay particular attention to allowing earcons to be clearly discernible while also maintaining musical

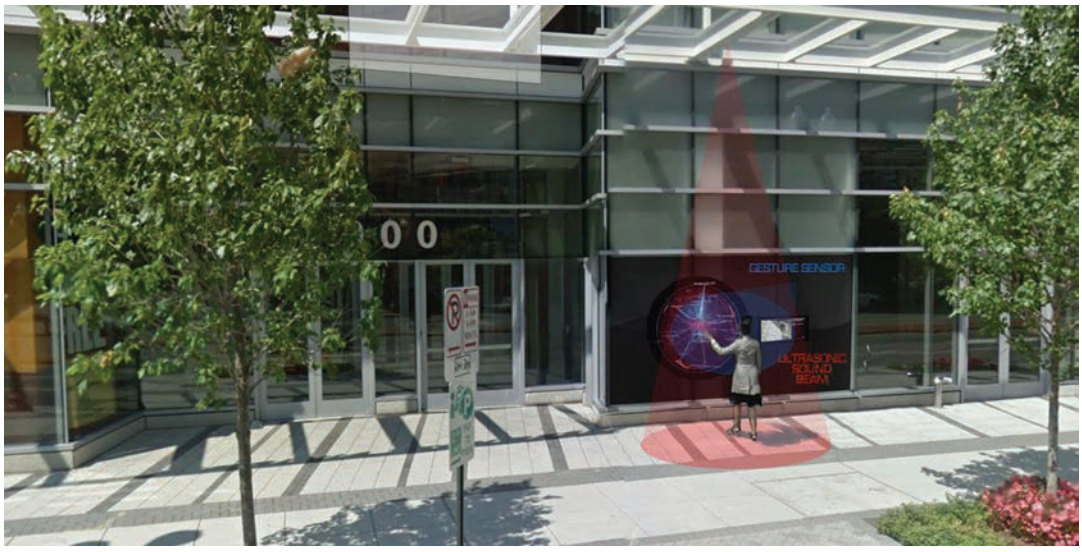


Figure 2. Rendition of the Virginia Tech National Capital Region (VTNCR). The kiosk-like implementation showcases an ultrasonic (holosonic) sound beam for localized audio and an embedded gesture sensor (street photo courtesy of Google Inc. Street View).

consonance with the underlying background music. While exploring the malleability of the aural stimuli, the team chose to use the Beads native Processing library,[3] because it greatly simplifies the network communication protocol. However, this means using only minimal algorithmic elements to provide variance—namely, an envelope ramp and randomly selected pregenerated subtle earcon variants.

The ensuing installation will be situated in the front window outside the VTNCR's main building in Arlington, Virginia (see Figure 2). We will use Kinect as a long-throw motion sensor for monitoring a passerby's motion throughout the width of the sidewalk, where observed kinetic energy rotates the Orb to attract participant attention. We envision projecting aural output using Holosonics' highly directional ultrasonic speaker array to limit environmental sound pollution and restrict the area within which the sound can be observed (see www.holosonics.com). The ultrasonic speaker array's output is only audible to the human ear, so after it refracts from a surface (a human body), the ensuing sound perception offers a uniquely surreal impression of someone speaking or whispering into one's ear.

We chose to leverage this effect to reinforce the intent of capturing a passerby's attention by directing a prerecorded voice to a listener, thus encouraging the person to step closer to the installation. Simultaneously, the display will show a pulsating outline of a human palm, encouraging potential participants to step closer within the range of a Leap Motion, whose primary purpose is to track arm motion as a means of primary interaction with the user interface.

Once a participant enters the Leap Motion controller's field of view, the Orb will wake from its dormant state, zoom out, and populate project nodes across the Orb's surface while also displaying possible actions in the top left corner of the screen.

The participant can navigate through the nodes of projects by rotating the X and Y axes (with Z being an alternative to the Y axis, depending on the orientation of the Leap Motion's mount point). By extending an index finger, a magnetic cursor appears on screen that can be navigated to any of the front-facing project nodes using X and Y axes (once again, with Z being an alternative to the Y axis, depending on the orientation of the Leap Motion's mount point). For the sake of making the navigation simpler, the cursor has a built-in gravitational pull toward the nearest project node.

Once a project is invoked (with a small delay used to highlight a rendering of new visual connections), the project overlay slides in from either the right or left (depending on where it was stowed the last time it was invoked) and the media (if any) commences playback. Within this mode, the default icons describing possible hand gestures (actions) are replaced with sliding

left or right action designed to stow the project description returning to the default navigation mode.

Following the stowing of the project description, users can return to the default top-level mode or continue to navigate through the tree of related objects using the Orb rotation and cursor navigation.

Through an iterative design process, we have worked through possible outlier or unusual occurrences, such as the user pointing his or her index finger with an extended thumb, which could cause problems with gesture detection. Minimizing potential misrepresentation of hand states will require additional filtering and the use of distinct gestures and hand positions that are significantly distinguishable from each other.

Generating the Database

Another critical aspect of the second phase is the work on the back end. Although we primarily generated the database during the first phase, it was OS-specific, lacked automation, and lacked a simple way for adding new entries. As a result, the second phase focuses on providing a Web-based interface for adding new content, including supporting media materials, as well as implementing and automating database generation in an OS-agnostic fashion.

For this purpose, we're using a combination of Python and Apache tools, generating a simple tab-delimited text file that references locally stored media materials. We've also adapted the Orb to handle up to 20 concurrent data nodes, always referencing the 20 newest projects and leveraging their relationships to older projects to let users, in due time, navigate through the entire tree, should they desire.

Although the current iteration of the Orb is designed to theoretically scale up to handle much larger datasets, it is not entirely clear whether the usefulness of its relational navigation would scale with the growing database. For us, the greatest lesson learned by embarking on this journey was ultimately not its big data potential. Rather, it is the unpredicted impact achieved by exploring cross-relations in a controlled small data setting. Our experimentation with a small dataset uncovered a disconnect between an organization's structure and the evolving mission and intent of the institute, ultimately revealing actionable insights and leading to concrete change. **MM**

The greatest lesson learned [had to do with] the unpredicted impact achieved by exploring cross-relations in a controlled small data setting.

Acknowledgments

The team would like to thank the Virginia Tech Institute for Creativity, Arts, and Technology, as well as Virginia Tech National Capital Region for their support of this project.

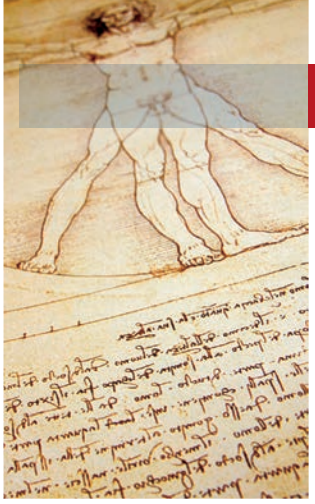
References

1. S.Y. Mousavi, R. Low, and J. Sweller, "Reducing Cognitive Load By Mixing Auditory and Visual Presentation Modes," *J. Educational Psychology*, vol. 87, no. 2, 1995, pp. 319–334.
2. M.M. Blattner, D.A. Sumikawa, and R.M. Greenberg, "Earcons and Icons: Their Structure and Common Design Principles," *Human–Computer Interaction*, vol. 4, no. 1, 1989, pp. 11–44.
3. E.X. Merz, *Sonifying Processing: The Beads Tutorial*, CreateSpace Independent Publishing Platform, 2011.

Dane Webster is an associate professor at Virginia Tech. Contact him at webster@vt.edu.

Ivica Ico Bukvic is an associate professor at Virginia Tech. Contact him at ico@vt.edu.

This article originally appeared in IEEE MultiMedia, *vol. 23, no. 1, 2016.*



Editors: **Robert R. Hoffman, Jeffrey M. Bradshaw, and Ken Ford,** Florida Institute for Human and Machine Cognition, rhoffman@ihmc.us

To Know or Not to Know, What Is the Need?

Robert R. Hoffman, *Institute for Human and Machine Cognition*
Matthieu Branlat, *361 Interactive*

> The old idea that we will only share information with someone who needs to know, that's sort of the tagline from movies and whatnot, is basically flawed because—who knows who needs to know? — Stanley McChrystal, US Army (Retired)[1]

A previous essay in this department discussed the implications of a "fundamental disconnect"[2]: that the time frame for effective experimentation to validate new intelligent systems technology is vastly outpaced by the time frame of change in technology. This essay deals with another fundamental disconnect related to intelligent systems technology, hinted at by this essay's title. In a recent essay on current challenges for the intelligence community, Josh Kerbel, the chief analytic methodologist for the Defense Intelligence Agency, said,[3]

> Analytic organizations need to be much flatter and more dynamically networked. The traditional fixed and compartmentalized hierarchies—often rooted in secrecy-driven compartmentalism, are not agile and impede holistic thinking…Analysts need to think of and use technology as a cognitive aid and not just as a tool for data management and communication…visualization is a crucial aid to thinking holistically and understanding complex issues… [also] analysts need to get past their "secrecy bias"—the notion that classified information is almost always better than open source. In an open world, this simply cannot remain a fundamental presence.

These comments eloquently express the themes of this essay.

The Need to (Not) Know

At any given time, there will be certain things that a given person does not need to know and that he or she must be prevented from knowing. Technically, the custodians of classified information must establish, prior to disclosure, that the intended recipient must have access to the information to perform his or her official duties.[4] This "need to know" tradition and policy is ensured and maintained through mechanisms of security clearance and compartmentalization. It is really a "need to not know" policy. Compartmentalization is with respect to types, sources, and topics of information, and the methods by which information is collected. Even if an individual has a high-level clearance with regard to a certain topic, he or she would be prohibited from knowing information that was not pertinent to the assigned duties and immediately necessary to fulfill specific obligations. For example, an individual might have a high-level clearance regarding information about cyberdefense, but would only be permitted to obtain information pertinent to his or her specialization in the security of the electric utilities. As another example, this same hypothetical individual might be able to obtain information about the most recent hacker activity aimed at some part of the nation's utilities infrastructure, but might not be able to learn how that information was obtained.

To Share, or Not to Share, That Is the Question

Cyber threats (such as identify theft, hacks, cyberattacks, and the use of Twitter to promote radicalism) have become clear and present in recent years. We have heard strident pleas that there needs to be more information sharing within and between organizations, with business and government sharing more information, and government agencies sharing more information. If only the FBI and CIA had shared information, it was

 Published by the IEEE Computer Society

said, the 9/11 attacks might have been averted. Government agencies have established various information-sharing portals. The Department of Homeland Defense was created to allow individuals who ordinarily would remain cloistered in individual acronyms (CIA, FBI, DoD) to work collaboratively. In 2004, FBI Deputy Assistant Director for Counterterrorism Willie Hulon testified before the House Government Reform Subcommittee on Technology, Informational Policy, Intergovernmental Relations and the Census, saying,[5]

> To achieve success in this war on terror, we have transformed the FBI's Counterterrorism Division…to one that is more collaborative and proactive…we have improved information sharing with other federal agencies and state and local law enforcement entities…A major element… is our increasing integration and coordination with our partners in the US and international law enforcement and intelligence communities. More than any other type of enforcement mission, counterterrorism requires the participation of every level of local, state, national, and international government.

The urgency of developing improved cyberdefense capabilities, on the part of businesses as well as governments, has been expressed pointedly across the US political landscape, including statements made by President Barack Obama. Executive Order 13636 says,[6]

> Repeated cyber intrusions into critical infrastructure demonstrate the need for improved cybersecurity. The cyber threat to critical infrastructure continues to grow and represents one of the most serious national security challenges we must confront. The national and economic security of the United States depends on the reliable functioning of the Nation's critical infrastructure in the face of such threats. It is the policy of the United States to enhance the security and resilience of the Nation's critical infrastructure and to maintain a cyber environment that encourages efficiency, innovation, and economic prosperity while promoting safety, security, business confidentiality, privacy, and civil liberties.

The Need to Share

Continuing from Executive Order 13636,[6]

> We can achieve [cyber security] goals through a partnership with the owners and operators of critical infrastructure… Policy coordination, guidance, dispute resolution, and periodic in-progress reviews for the functions and programs described and assigned herein shall be provided through [an] interagency process…It is the policy of the United States Government to increase the volume, timeliness, and quality of cyber threat information shared with U.S. private sector entities so that these entities may better protect and defend themselves against cyber threats…The Secretary [of Homeland Security] and the Attorney General…shall establish a process that rapidly disseminates the reports… Such process shall also, consistent with the need to protect national security information, include the dissemination of classified reports to critical infrastructure entities authorized to receive them… the Secretary [of Homeland Security]… in collaboration with the Secretary of Defense, shall…establish procedures to expand the Enhanced Cybersecurity Services program to all critical infrastructure sectors.

In other words, we have to share. The executive order continues,[6]

> The Secretary [of Homeland Security]… shall expedite the processing of security clearances to appropriate personnel employed by critical infrastructure owners and operators, prioritizing the critical infrastructure…the Secretary [of Homeland Security] shall expand the use of programs that bring private sector subject-matter experts into Federal service on a temporary basis. These subject matter experts should provide advice regarding the content, structure, and types of information most useful to critical infrastructure owners and operators in reducing and mitigating cyber risks.

Again, we have to share. But there is an important caveat in Executive Order 13636:[6]

> This voluntary information sharing program will provide classified cyber threat and technical information from the Government to eligible critical infrastructure companies or commercial service providers that offer security services to critical infrastructure.

The key words here are "classified" and "eligible." We have to share, but we must maintain the Need to (Not) Know policy. Thus, we seem to have a paradox on our hands.

The Real Need (Is to Know)

In the current climate and situation, Need to (Not) Know is a recipe for disaster, if not just failure. Thus, we propose the New Need to Know principle:

> At any given time, any given person might need to know anything and must be supported in finding it.

This principle accords with HCC and, in particular, the Sacagawea Principle discussed in a previous essay, which states,[7]

> Human-centered computational tools need to support active organization of

> information, active search for information, active exploration of information, reflection on the meaning of information, and evaluation and choice among action sequence alternatives.

This principle was in reference to work system design, and displays in particular, but it is a broad statement proclaiming the necessity, and not just the importance, of human sensemaking.

Such a proclamation notwithstanding, the Need to (Not) Know policy and its associated laws and mechanisms are not likely to change anytime soon. Scientists might like to engage in research on, say, cyberdefense or the development of software tools to assist intelligence analysts, or the creation of new visualizations for big data, but if they do not already have clearance, they can look forward to at least a six-month delay in obtaining even a minimum clearance. And that would be after additional months of delay in navigating the bureaucratic machinery needed to get the gears of the clearance approval process moving in the first place. The primary means that organizations exercise for coping with this is to anticipate clearance issues by hiring people who are eligible for clearance, but many projects cannot exercise this option. In general, there is a lack of flexibility due to time constraints in bringing in new resources. For ourselves, we have seen many bright and enthused young computer scientists and cognitive systems engineers express interest in helping to solve the urgent national and international problems and work on the core issues to develop scientific solutions—but they can't.

Getting around the Paradox

What is being done about the paradox? Mostly it is being ignored, and people continue to just muddle through. The clearance bottleneck is routinely bemoaned when shorter-term projects are designed and stood up. Looking back over the past 10 or so years, we have seen many Small Business Innovation Research (SBIR) opportunities in which the clearance bottleneck was a potential showstopper. By the time you get the clearance, the final report is due.

Because it is difficult for an individual or an organization (especially a small business or researchers in academia) to hold all the potential levels and types of clearance necessary in the compartmentalized world of security, a potential approach is to rely on the constitution of the work team. Large organizations might be able to cover the various types of requirements internally, whereas smaller ones might rely on project-based collaborations with other organizations. In any case, the common situation is that not everybody in the research team holds all or even most of the necessary credentials. By itself, this situation hinders communication and collaboration concerning the specifics of the systems or events under investigation that fall under the Need to Know framework. As a consequence, for collaboration to be meaningful, the team members who are allowed to access information need to be able to abstract away and describe to others the nature of the situations at hand. From our experience working in such teams with diverse levels or types of clearance, this is a difficult task. In addition to uncertainties related to what specifics can be discussed, and how, such exchange of information requires that people "in the know" understand what kind of information might be valuable to others. This is possible only if there is common ground about each other's interests and expertise—that is, it is effective only in teams of people who have sufficient overlap and experience working with each other.

There have been grumblings of discontent about Need to (Not) Know in the acronymmed hallways. The concept of a New Need to Know principle has been raised explicitly at recent meetings on cyberdefense and intelligence analysis. When the New Need to Know postulate was formulated more than 10 years ago (by the senior author of this essay), it was scary because it advocated a taboo and was proposed by an "outsider." Now, it is not so scary—and it has to be on the table.

Solution #1: Reality Checkers

One potential way forward is suggested by another statement in Executive Order 13636.[6]

> In order to maximize the utility of cyber threat information sharing with the private sector, the Secretary [of Homeland Security] shall expand the use of programs that bring private sector subject-matter experts into Federal service on a temporary basis. These subject matter experts should provide advice regarding the content, structure, and types of information most useful to critical infrastructure owners and operators in reducing and mitigating cyber risks.

Some years ago, cognitive systems engineers proposed the establishment of a group called the Friends of the Intelligence Community. This collective was motivated by the recognition that private sector vendors were creating software decision support and visualization tools intended for use in defense and in intelligence analysis, and thrust into the workplace, but which were not human-centered in their design. The software tools and interfaces were based on a designer-centered philosophy rather than a robust cognitive work analysis. Not

surprisingly, many of the software tools ended up collecting dust.[8,9] The Friends offered themselves as fair witnesses who would comment on emergent procurements and offer independent evaluations. Such a solution corresponds to strategies used by large government contracting companies that recruit retired personnel from military organizations, thus constituting pools of in-house domain experts. The Friends would have extended such a strategy to a larger community of interest and might have entrained connections inspiring new research opportunities.

Something like the Friends, but turned on its head, might be one way to deal with the paradox of the Need to (Not) Know versus the New Need to Know. Specifically, retiring experts in cyberdefense would make themselves available on an ad hoc basis to provide reality checks to computer scientists and cognitive systems engineers on their notions and designs for intelligent systems that are intended to assist in cyberdefense operations. Because many analysts and cyber workers who retire from government service go into lucrative consultations, their service as reality checkers would have to be incentivized, even though their service would hinge on intrinsic motivation. Certainly, a single reality check consultation would not require much time. Even at a standard government rate for senior scientists, the net sum would be a rounding error at the third decimal point compared to most major defense acquisition projects.

Such an institution of reality checkers would require mechanisms and procedures, of course, but these need not and should not be detailed prematurely. It is, however, important to note that the institution could serve multiple purposes. It would help the first generation of cyber defenders pass their knowledge on to the second generation and stay engaged after their retirement. And it might help a generation of technologists build genuinely usable and useful intelligent systems. Furthermore, creating a community of reality checkers might lead to enhanced common ground between the operational and R&D communities. Such common ground would facilitate exchanges in particular by allowing reality checkers to understand what matters for R&D and thus provide more targeted information.

Solution #2: Private Sector Analogs

Many work situations are not observable directly: when cognitive systems engineers studied firefighters, they were not, for obvious safety reasons, inside the buildings shadowing the firemen. Different cognitive task analysis techniques let researchers gather data about work situations without relying on direct observation. Another approach is to use analogs—that is, to rely on work situations that present similar characteristics despite differing on specifics. Such an approach allows for the study of the problem of interest while working around safety and security constraints. A recent study examined cybersecurity in the context of the energy sector.[10] Although cyber operations conducted on the military side differ, they face some of the same fundamental problems as those associated with securing a private company or organization network. When appropriate, similar activities in the private sector provide relevant analogs. Another example is the use of business intelligence as an analogue to open source intelligence.

Two main issues are associated with using analogs. The first relates to the risk of missing significant differences between two seemingly equivalent domains (where specifics might create fundamentally different situations). Domain experts such as the reality checkers discussed earlier would constitute an effective source for insights into analogs' limitations—they would be able to point out differences to consider in order to investigate the problem in a meaningful way. Additionally, organizations funding the research must agree on the relevance of using analogs in lieu of the situations they are specifically intending to support. This issue requires researchers to be able to convince those organizations, a process that can be difficult in more exploratory contexts (for example, when researchers are interested in finding out whether a situation can serve as a relevant analogue to the one under investigation).

When it comes to designing technologies to improve a cognitive work domain, there is no substitute for first learning how to do the work.[11] But in the case of cyberdefense, the difficulty of overcoming the clearance hurdle mandates that we come up with alternative solutions, any solutions that might help, whether in small ways or large ways.

There is great benefit to keeping research at an unclassified level in order to be able to share insights and results outside of the project team. R&D progresses in a large part through sharing, via publications and communications. That being said, the New Need to Know principle is not actionable within the operational cyberdefense community. But, as we argue in this essay, it does encourage consideration of possibilities for advancing cyberdefense intelligent systems. ■

References

1. S. McChrystal, interview by K. Ryssdal, *Marketplace*, Nat'l Public Radio, 11 May 2015.

2. P.S. Hancock and R.R. Hoffman, "Keeping Up with Intelligent Technology," *IEEE Intelligent Systems*, Jan./Feb. 2015, pp. 62–65.
3. J. Kerbel, "The U.S. Intelligence Community's Kodak Moment," *Nat'l Interest*, 15 May 2014, p. 2.
4. *Department of Defense Dictionary of Military and Associated Terms*, US Gov't Printing Office, 2015.
5. W.T. Hulon, *Statement to the House Government Reform Subcommittee on Technology, Informational Policy, Intergovernmental Relations and the Census*, 13 July 2004.
6. Executive Order 13636, *Improving Critical Infrastructure Cybersecurity*, 12 Feb. 2013.
7. M.R. Endsley and R. Hoffman, "The Sacagawea Principle," *IEEE Intelligent Systems*, Nov./Dec. 2002, pp. 80–85.
8. R.R. Hoffman and W.C. Elm, "HCC Implications for the Procurement Process," *IEEE Intelligent Systems*, Jan./Feb. 2006, pp. 74–81.
9. K. Neville et al., "The Procurement Woes Revisited," *IEEE Intelligent Systems*, Jan./Feb. 2008, pp. 72–75.
10. M. Branlat, A.M. Morison, and D.D. Woods, "Challenges in Managing Uncertainty During Cyber Events: Lessons from the Staged-World Study of a Large-Scale Adversarial Cyber Security Exercise," *Proc. ASNE Human Systems Integration Symp.*, 2011, pp. 10–25.
11. K.B. Bennett and R.R. Hoffman, "Principles for Interaction Design, Part 3: Spanning the Creativity Gap," *IEEE Intelligent Systems*, Nov./Dec. 2015, pp. 82–91.

Robert R. Hoffman is a senior research scientist at the Institute for Human and Machine Cognition. Contact him at rhoffman@ihmc.us.

Matthieu Branlat is a research scientist at 361 Interactive. Contact him at matt@361interactive.com.

This article originally appeared in IEEE Intelligent Systems, *vol. 31, no. 1, 2016.*

Editor: **Rafael Prikladnicki**
Pontifica Universidade Catolica
do Rio Grande do Sul
rafael.prikladnicki@pucrs.br

Is an Athletic Approach the Future of Software Engineering Education?

Emily Hill, Philip M. Johnson, and Daniel Port

IN THE PAST 10 YEARS, there has been considerable evidence of the harmful effects of multitasking and other distractions on learning. One study found that multitasking students spend only 65 percent of their time actively learning, take longer to complete assignments, make more mistakes, are less able to remember material later, and show less ability to generalize the information they learned for use in other contexts.[1]

Traditional software engineering education approaches—in-class lectures, unsupervised homework assignments, and occasional projects—create many opportunities for distraction.

To address this problem, coauthor Philip M. Johnson developed an "athletic" software engineering education approach, which coauthors Emily Hill and Daniel Port adapted for use in their courses. We wanted to determine if software engineering education could be redesigned to be like an athletic endeavor and whether this would improve learning.

Athletic Software Engineering

We wanted to design the educational process to incentivize students to avoid multitasking and focus on learning complex, multistep tasks.

Athletic software engineering education adopts simple features of conventional athletic training. The primary goal is to minimize the time students need to accomplish a task. Many sports, such as running and cycling, are based on completing a task in a minimal amount of time. Another goal is to encourage a high-quality effort, which leads to better results.

Generally, neither feature is found in the software engineering classroom. Assignments usually eliminate time constraints. For example, if instructors believe a problem could be completed in a day, they might provide a week, thereby preventing students from claiming that they didn't have enough time to finish. Also, in software engineering, working quickly is typically viewed as working sloppily. This contrasts with athletic endeavors, in which sloppiness often produces slowness.

Athletic software engineering education resolves this dichotomy by differentiating between the creative aspects—for which minimum times can't be defined—and the mechanics—for which they can—of each skill to be taught.

Let's use writing a unit test as a simple example. In a lecture-based survey course, students might read a chapter about unit testing and learn how to compare and contrast it with integration testing, load testing, and other kinds of testing. The instructor might require students to express this conceptual knowledge via

2376-113X/16/$33.00 © 2016 IEEE Published by the IEEE Computer Society

a written exam. In a project-based practicum, students might have to develop unit tests for an application. Different groups might develop their tests at different times and with different technologies. In a flipped classroom, students might learn about unit testing at home via videos and develop unit tests in class under the instructor's guidance.

In the athletic approach, unit-test writing combines creative decisions (deciding what to test and why) and mechanics (performing the tasks necessary to yield high-quality software).

The mechanics of developing even a simple unit test involve multiple languages, tools, and technologies. Students can be incapable of developing unit tests or take a lot of time to do so not because of their creative decisions but because they haven't mastered the mechanics. The good news is that by integrating athletic concepts into the curriculum, students can master these mechanics without experiencing distractions.

In a nutshell, athletic software engineering education involves

- structuring the curriculum as a sequence of skills to master, not concepts to memorize;
- creating a set of training problems for each skill, accompanied by a video demonstrating how to solve them in a minimal amount of time;
- providing the opportunity to learn to solve the problems in the prescribed amount of time;
- testing mastery of a skill through an in-class, timed problem, similar to physical training's workout of the day (WOD); and
- acquiring the next skill, typically by employing many of the tools and technologies previously learned.

> Students agreed that the athletic software engineering education approach kept them focused.

The website for Johnson's Spring 2015 advanced software engineering class at the University of Hawaii at Manoa (http://philipmjohnson.github.io/ics613s15) provides a complete example of applying athletic software engineering to a variety of skills.

This approach requires students to demonstrate mastery of various software engineering skill sets' mechanics via assessments that they must complete correctly within a time limit. This reduces distraction, improves focus, and makes learning more efficient.

Evidence

The athletic approach has been evaluated in two software engineering courses by Johnson, adapted to a business-school curriculum by Port, and adapted to an elementary programming class by Hill.

Athletic Education in Software Engineering

Johnson used an athletic style to teach software engineering to an undergraduate software engineering class in 2014 and a graduate software engineering class in 2015. The two had a total of 29 students. To assess the approach, he required students to write technical essays on their progress and administered a questionnaire near the semester's end that obtained their opinions.

Of the students surveyed, all but one (97 percent) preferred the athletic course structure to the traditional one. A participant commented,

> *I would choose to do [academic] WODs over the traditional approach because it helps you to become accustomed to working under pressure. I find myself learning more this way due to having to remember what I've done rather than searching for how to do something and then forgetting soon after.*

Athletic software engineering lets students repeat training problems if they don't achieve adequate performance. In our study, 72 percent of them found it useful to repeat the problems, and most repeated more than half of the problems at least once.

Of responding students, 82 percent said athletic software engineering improved their focus while they learned the material. One commented,

> *Like many students, when I do work at home, I get distracted easily. … WODs definitely helped me to accomplish more in less time.*

Pressure is a part of a software developer's life. More than 80 percent

of the students said the athletic approach helped them feel comfortable programming under pressure.

Athletic Education in Business School

Port adapted the athletic approach to an introductory Web-application-programming course for management of information systems (MIS) majors. The challenge was to give novices basic programming fluency, skills and strategies for becoming efficient in all software development phases, and an understanding of why and where MIS workers use these abilities. We wanted to use the athletic approach to rapidly build competence and confidence in developing software to improve students' future performance in MIS courses.

Our experience over the past year indicates the athletic approach was highly effective in achieving these goals. Unexpectedly, it also generated enjoyment and enthusiasm for building software once the students achieved competence and confidence. In addition, it fostered both the determination to make software work and elation when it did, rather than fear and sadness when it didn't. Port's students said that the athletic approach promoted greater collaboration and that they didn't feel competition but instead wanted to help one another understand the material and master the assignments.

Students liked the practice WODs and learned a great deal by trying them and then watching a video of the solution. However, they didn't like in-class WODs and were frustrated when they repeatedly didn't finish them. Nevertheless, they eventually succeeded and decided that WODs were essential for building programming competence. Running WODs until students could finish them built confidence and enthusiasm. Upon completion, students felt ready to take on the challenge of building full applications with more complexity and less guidance.

Students who experienced the athletic approach did better than those whose classes took a more traditional approach, and a higher percentage performed successfully in subsequent MIS courses that depended on development skills. However, the athletic approach discouraged some students who didn't do as well as they expected or who weren't as successful as other students.

Athletic Education in Introductory Programming

Hill adapted the athletic approach for introductory programming classes in Python and Java. She assigned the in-class, timed problems as homework if the students didn't finish. However, to receive an A on an assignment, they had to correctly complete it in class.

Students said they liked working on the practice WODs and learning from the videos, and sometimes requested more of each to help learn difficult concepts.

Anonymous student survey feedback was mixed. In the Python course, 18 of the 25 students responded, with two-thirds preferring the athletic approach over a more traditional style. Unfortunately, in the Java course, only five of 24 students responded, rendering the results insignificant. Unlike Port's students, those in Hill's Python and Java classes complained that the WODs' competitive nature discouraged collaborative learning. For example, one said,

> *[I]t created a hostile environment where people were afraid to admit that they didn't understand course material outside of class. Also, it made peers less likely to help each other or provide advice.*

On the other hand, another student noted that the competition spurred them to "do additional work using resources outside of the class."

Both courses' students agreed that the athletic structure kept them focused and that they really liked the practice WODs. Said one,

> *It was less stressful doing [practice WODs] because I knew that the homework was not graded. The homework was there solely to help me learn, and that absence of negative pressure allowed me to focus and concentrate more than I usually do.*

> Traditional software engineering education approaches create many opportunities for distraction.

Based upon our initial experiences, we believe an athletic pedagogy will find its place as a way to help students efficiently master software engineering's

mechanics and better enable them to handle the creative problem solving that our discipline requires. As the diverse student responses to different adaptations showed, the approach is still in its infancy. We will continue to refine and improve it with additional experience and invite software engineering educators who find this approach of interest to join us.

Reference

1. A. Murphy Paul, "How Does Multitasking Change the Way Kids Learn?" *MindShift*, 3 May 2013; http://ww2.kqed.org/mindshift/2013/05/03/how-does-multitasking-change-the-way-kids-learn.

EMILY HILL is an assistant professor of computer science in Drew University's Department of Mathematics and Computer Science. Contact her at emhill@drew.edu.

PHILIP M. JOHNSON is a professor in and the associate chair of the University of Hawaii at Manoa's Department of Information and Computer Sciences. Contact him at johnson@hawaii.edu.

DANIEL PORT is an associate professor in the University of Hawaii at Manoa's Information Technology Management Department. Contact him at dport@hawaii.edu.

This article originally appeared in IEEE Software, *vol. 33, no. 1, 2016.*

Descaling Your Scrum

James O. Coplien

The waterfall software development process seemed to be well-suited to developing simple or perhaps complicated systems. Such systems can be master-planned from the outset, with work done by specialized teams. Waterfall has feedback but success does not depend on efficient feedback, because the process presumed to foresee the next steps. Because its stages (analysis, design, implementation, testing) were held to be largely independent and simple, the work was partitionable. You could scale each phase independently to achieve maximum throughput.

Scrum is optimized for complex, adaptive systems. In the early days of our Scrum patterns effort, we included some scaling patterns. Jeff Sutherland rebuked us with the reminder that Scrum is fractal in nature: that is, it's a scale-free system. It grows the way an ecosystem grows: by local adaptation and piecemeal growth. It comprises cross-functional teams rather than independent (scalable) teams. Specialization doesn't handle this kind of complexity well because a problem in coding in this minute may require testing insight in the next minute and analysis insight two minutes hence.

A Scrum view envisions growth differently. You can grow a team's learning without adding people—the mind, at this level, is unlimited. Learning in turn increases throughput (better Kaizen) and, to quip a well-known book title, allows twice the work in half the time. In terms of human mass, Scrum organizations grow not by simple aggregation but by differentiation—the way an embryo develops. It is this differentiation rather than any notion of "scaling" that should be the business focus—unless you're building pyramids. That means thinking in terms of federations and partnerships instead of armies.

The goal isn't to grow one's organization, but rather to generate as much value with as few people as you can. Alex Lope-Bello, CEO of Comtrade, says, "Grow your business—not your teams." Scrum pundits talk about increasing teams' capacity to do work (called their velocity) by large integral factors or even orders of magnitude. If you can improve the process to realize a ten-fold gain, why would you hire ten times as many people instead? Twice the work in half the time.

Management feels safety in numbers, so descaling takes more courage than scaling. Let's say

2376-113X/16/$33.00 © 2016 IEEE Published by the IEEE Computer Society

that you have seen the light of agile and want to convert your organization. If you have a 70-person development organization, you might be tempted to teach Scrum to everyone on Thursday and Friday and to turn on the Scrum switch Monday morning. First, this is likely to be painful in the sense that Scrum will make it visible that you don't need many of the roles in your current organization. Second, it locks the organization into a local optimum from which it is unlikely to escape. Instead, start over with ten people. If you're courageous, you'll start with only five.

Start by cutting people; continue by shortening the work week. Sutherland relates a story about Scott Maxwell at OpenView Venture Partners. Maxwell noticed that increased hours at the office decreased output. Sutherland relates: "The peak of productivity actually falls at just under 40 hours a week. Armed with this data, Scott started to send people home early. 'It took them a while to get that I was serious,' Maxwell says." (Sutherland, "The Art of Doing Twice the Work in Half the Time")

How about the rest of the workers? Have them innovate new features or new products by building prototypes or talking with end-users. Have them start up a new product or a new business within the company. Grow the business—not the team.

Selected CS articles and columns are also available for free at http://ComputingNow.computer.org.

Finding the Cybersecurity Job You Want

Rakesh Verma is a professor of computer science at the University of Houston and director of its ReDAS (Reasoning and Data Analytics for Security) Laboratory. His research interests are formal methods and data analytics applied to natural-language understanding and cybersecurity. Thus, he is very knowledgeable about security-related careers. We asked Verma several questions about career opportunities in this increasingly important field.

ComputingEdge: What careers in computing technology will see the most growth in the next several years, and why?

Verma: I believe that cybersecurity careers will probably see the most growth and will be particularly good choices for data science/data analytics students in computer science. The reason is that the amount of data is growing exponentially and there is no way for organizations to analyze it all manually for cyberthreats. Another area that shows high potential is digital criminal forensics because digital and Internet espionage and cybercrime are on the upswing.

ComputingEdge: What would you tell college students to give them an advantage over the competition?

Verma: I would advise CS students to take data analytics—data mining, machine learning, natural language processing—and cybersecurity classes. I think these courses will make them stand out from the competition.

ComputingEdge: What advice would you give people changing careers midstream?

Verma: For those changing careers, my advice would be to get into cybersecurity. The more digital the society, the higher the demand for cybersecurity professionals. Besides digital and Internet espionage and crime, there could also be digital wars in the future between nation–states.

ComputingEdge: What should applicants keep in mind when applying for computer and cybersecurity jobs?

Verma: Applicants should remember that they must have good conceptual knowledge that is generalizable—since the computer field and cyberthreats are constantly evolving—as well as some solid practical skills so that they can hit the ground running.

ComputingEdge's Lori Cameron interviewed Verma for this article. Contact her at l.cameron@computer.org if you would like to contribute to a future *ComputingEdge* article on computing careers. Contact Verma at rmverma@cs.uh.edu.

2376-113X/16/$33.00 © 2016 IEEE Published by the IEEE Computer Society

PROGRAMMER ANALYST: Analyze, design, develop, test & implement Web based applications and Windows based applications using knowledge of C#.NET, VS 2010, XML, HTML, ASP.NET, VB.NET, WCF, Oracle 11g,SQL Server 2008, Javascript & Windows 98/00/03/NT. Must be willing to travel & reloc to unanticipated client locations throughout the US. Req MS comp sci, eng or rel. Mail resumes to Virtuosa Global Sols LLC 674 US Highway 202/206,Suite 4, Bridgewater,NJ 08807.

SOFTWARE ENGINEER - design, develop, test & implement application s/w utilizing knowledge of HP LoadRunner, HP Performance Center, Jira, Confluence, QTP/UFT, C, VB Script, Java, .NET, JavaScript, Oracle 10g/9i/8i, CA Wily Introscope 9.2, HP Sitescope, HP Quality Center/HP ALM, ITKO LISA, HP ALM, SQL and PL/SQL. Must be willing to travel & reloc to unanticipated client locations throughout the US. Reqs MS in comp sci, eng or rel. Mail resumes to Strategic Resources International, Inc. 777 Washington Rd, Suite 2, Parlin, NJ 08859

VLOCITY is seeking a Software Engineer in San Francisco, CA to design and implement product requirements that are usable, scalable, extensible, and maintainable. Ref Job ID: 9QE2NY & send res. to T. Dilley at hiring@vlocity.com

NETWORK SYSTEMS ADMINISTRATOR IN SKOKIE, IL. Responsible for dsgng, installing, & supporting the company's network & computer systems. Responsible for WANs, LANs, the internet & network segments & ensures system efficiency. Reqrd: Bachelor Deg or foreign equiv in Comp Info. Systems or Comp Sci., & 2 yrs exp in job offd. Must have exp working in a medical/home healthcare service envrmt performing database maintenance, systems integration & mgmt info. systems, including ability to use dsgn & dvlpmt skills (e.g. CSS, HTML), & mgmt & support of the WANs, LANs, internet, & network segments. Ability to work on problems of diverse scope in network & systems administration including mgmt, human resources, government regulations, planning, & dvlpmt, according to our customers' needs. Mail resumes to Fazlur Rahman, President, Home and Wound Care Physicians, PC, 8328 Lincoln Ave., Skokie, IL 60077. Ref. No. 814407392D; no calls, emails or fax.

ENGINEER: Broadcom Corporation, the leading provider of highly integrated complete system-on-a-chip solutions for digital and satellite cable set-top boxes, cable and DSL modems, residential gateways, high speed transmission area networking, home and wireless networking, cellular and terrestrial wireless communications, VoIP gateway and telephony systems, broadband network

INFOSYS LIMITED has multiple, full-time openings in Plano, TX and various and unanticipated locations throughout the U.S. Must be willing to work anywhere in the U.S. as the position may involve relocation to various and unanticipated client site locations; any relocation to be paid by employer pursuant to internal policy. Equal Opportunity Employer M/F/D/V. Please apply to Infosys Limited online at: https://www.infosys.com/careers/job-opportunities/pages/index.aspx. Select the green button "Search for Jobs in Americas" which will bring you to the site for Experienced Professionals. Once a user account has been created, please follow the link for 'Search Openings' and enter the job # listed below in the 'Auto Req ID' box."

SENIOR PROJECT MANAGER(S) - U.S. needed in Plano, Texas, and various and unanticipated locations throughout the U.S. to lead medium- to large-scale IT engagements with responsibility for drafting proposals, schedules, and cost estimates, performing requirements analysis, defining architecture, and testing and implementing IT solutions. **(JOB # 12949BR).**

PROJECT MANAGER(S) - U.S. needed in Plano, Texas, and various and unanticipated locations throughout the U.S. to help gather requirements, define architecture, and determine scope to deliver IT solutions. **(JOB # 12953BR).**

PROJECT MANAGER(S) - U.S. (Testing) needed in Plano, Texas, and various and unanticipated locations throughout the U.S. to perform activities to ensure that quality software work products are delivered on schedule, including coordination with clients and internal teams across the globe. **(JOB # 12952BR).**

TECHNOLOGY LEAD(S) - U.S. (Enterprise Solutions) needed in Plano, Texas, and various and unanticipated locations throughout the U.S. to design, develop, test and deploy specific modules for software products. **(JOB # 12958BR).**

TECHNICAL TEST LEAD(S) - U.S. needed in Plano, Texas, and various and unanticipated locations throughout the U.S. to test assigned modules for software products. **(JOB # 12954BR).**

TECHNOLOGY LEAD(S) - U.S. (Infrastructure Management) needed in Plano, Texas, and various and unanticipated locations throughout the U.S. to design, develop and deploy IT solutions for infrastructure environments, including evaluation of OS, DB, storage, network enterprise applications and middleware. **(JOB # 12959BR).**

TECHNOLOGY LEAD(S) - U.S. (Open Systems) needed in Plano, Texas, and various and unanticipated locations throughout the U.S. to design, develop, test, and deploy specific modules for software products. **(JOB # 12961BR).**

TECHNOLOGY LEAD(S) - U.S. (Mainframe) needed in Plano, Texas, and various and unanticipated locations throughout the U.S. to design, develop, test, and deploy specific modules for software products. **(JOB # 12960BR).**

TECHNOLOGY LEAD(S) - U.S. (Engineering) needed in Plano, Texas, and various and unanticipated locations throughout the U.S. to design, develop, test, and deploy specific modules for software products. **(JOB # 12957BR).**

TECHNOLOGY LEAD(S) - U.S. (Business Intelligence) needed in Plano, Texas, and various and unanticipated locations throughout the U.S. to design, develop, test and deploy specific modules for software products. **(JOB # 12955BR).**

TECHNOLOGY LEAD(S) - U.S. (Enterprise Application Integration) needed in Plano, Texas and various and unanticipated locations throughout the U.S. to design, develop, test, and deploy specific modules for software products. **(JOB # 12956BR).**

 Published by the IEEE Computer Society 2376-113X/16/$33.00 © 2016 IEEE

processors, and server solutions seeks all levels of Engineers/Scientists in Irvine, CA, Santa Clara, CA, San Jose, CA, Sunnyvale, CA San Diego, CA, Matawan, NJ, Chandler, AZ, Duluth, GA, Andover, MA, Edina, MA, Bellevue, WA, Horsham, PA, Austin, TX, Fort Collins, CO, Durham, NC, and Federal Way, WA: Test (ENG472), Product Engineering (ENG476), RF/Wireless (ENG484), Hardware Development (ENG502), Electronic Design (ENG506), IC Design (ENG507), Software Development (ENG510), Software Applications (ENG514), Software Systems (ENG516), Software Quality Assurance (ENG518), Systems Design (ENG520), Firmware (ENG521), Product Applications/Systems Integration (ENG584), DSP (ENG509), CAD (ENG 511), Configuration/Release (ENG519), Packaging (ENG524), Design (ENG537), Field Applications (ENGS87), Layout Design (ENG0313), Product/Prod. Line Manager (ENG586), Systems/Database Administrator (ENG641), Applications Programmer (ENG6555), and Process Development (ENG526), Sales/Business Development (ENG567), IT Security Analyst (ENG658); Business Systems Analyst (ENG646); Product Development (ENG558); Engineering Systems Analyst (ENG533) Compliance Engineer (ENG463); and Manager Enterprise Application (ENG347) (Oracle, SAP, Baan, etc). Education/experience requirements vary by position/level. Some positions may require domestic and/or international travel. Must have unrestricted right to work in U.S. Mail all resumes to HR Ops Specialist, 5300 California Avenue, Bldg. 2 #22108-B, Irvine, CA 92617. Must reference job code.

ERICSSON INC. has openings for positions of: **TECHNICAL SUPPORT ENGINEER _ in ATLANTA, GA** to perform as domain lead for SAP application Provide operational Tier-2 support to customers. Job ID: 15-GA-3583. **CONSULTING MANAGER _ in ATLANTA, GA** to provide leadership for the management & delivery of strategy projects for Ericsson & external clients by integrating telecom, media, & technology industry expertise & knowledge related to business operations. Job ID: 15-GA-1703. **ENGINEER – QUALITY ASSURANCE _ in MANCHESTER, NH** to perform requirement reviews, all phases of software testing, metrics tracking, & technical feature ownership. Job ID:

JUNIPER NETWORKS

Juniper Networks is recruiting for our Sunnyvale, CA office:

Software Engineer Staff #16072: Design, develop troubleshoot, and sustain packet forwarding engine and its associated software in highly complex, high performance networks.

Resident Engineer Staff #6660: Design, develop, and troubleshoot networks and provide operations support along with network and configuration analysis.

Resident Engineer Staff #24084: Design, develop and implement hardware and software certification testing and troubleshooting systems for company products. Prepare individual test cases incorporating the customer's proprietary network design elements, diagnostic programs, test fixtures, and equipment, for company's routing, switching, and security products against customized plans. May work at other undetermined locations throughout the U.S.

Software Engineer #19001: Design, develop, troubleshoot and debug mobility packet core gateway protocols, features, and infrastructure software support on company's MX Edge router series/distributed platform.

Mail single-sided resume with job code # to
Juniper Networks
Attn: MS A.8.429A
1133 Innovation Way
Sunnyvale, CA 94089

ARROWSIGHT INC.,
TEAM LEAD VIDEO BROWSING SOFTWARE.

Opening at Arrowsight, Inc, advanced technology surveillance firm in Mt. Kisco, NY for Team Lead, Video Browsing Software to: 1) Lead a team of software engineers implementing new components for browsing compressed video, using Agile / Scrum techniques. 2) Architect new software components, written in a C++-family object-oriented language, that analyze bitstreams in H.264 format, selecting video frames of interest, and assembling, tagging, and caching subsequences of these video and audio bitstreams for the purpose of browsing compressed video, using 3rd-party video decompression libraries as a foundation. 3) Review in detail and debug existing C++ video tools based on Microsoft DirectShow, ActiveX, Microsoft's Component Object Model (COM), and the Intel Integrated Performance Primitives library, including focus on interoperability mechanisms and memory management between C++, COM, and Microsoft CLR (Common Language Runtime). 4) Propose, architect, and implement extensions to video tools, written in C++-family object-oriented languages, adding multi-threaded code to retrieve and play multiple video and audio streams simultaneously and synchronously ensuring robust response to variations in frame durations. 5) Design and implement new multi-scale change-detection algorithms, written in C++-family object-oriented languages, using hardware-accelerated multi-stream image filtering. 6) Analyze recorded bitstreams from surveillance cameras that claim to generate H.264 or MPEG4-compliant bitstreams to determine compliance and features used or omitted. 7) Lead reviews of multithreaded C++-family object-oriented code written by colleagues and subordinates to identify bugs and ensure efficiency, and provide documentation for new and modified software. 8) Deliver new and modified code using SVN revision control software. 9) Supervise weekly testing of video tools and web site for individual software changes and during quality assurance phase of release cycle. Work with remote software customers to test new releases in their environments. Requires Master's degree in Computer Science or Computer Engineering and two years' experience. Forward resume to SWEng Hiring - Code CC16, Arrowsight, Inc. 45 Kensico Drive, Mt. Kisco, NY 10549 or via email with Code CC16 in subject line to jobs.cc16@arrowsight.com

INFOSYS LIMITED is in need of individuals to work full-time in Plano, Texas and various and unanticipated locations throughout the U.S. Must be willing to work anywhere in the U.S. as all job opportunities may involve relocation to various and unanticipated client site locations; any relocation to be paid by employer pursuant to internal policy. We have multiple openings for each job opportunity, and are an Equal Opportunity Employer M/F/D/V. Please apply to Infosys Limited online at: https://www.infosys.com/careers/job-opportunities/pages/index.aspx. Select the green button "Search for Jobs in Americas" which will bring you to the site for Experienced Professionals. Once a user account has been created, please follow the link for 'Search Openings' and enter reference ID(s) for the position(s) of interest in the 'Auto Req ID' box.

ASSOCIATE ENGAGEMENT MANAGER(S) needed to contribute to IT competitor analysis and prospect identification; provide ground intelligence to pursuit teams, as well as account context and client introductions required for opening diverse service offerings in account(s). Travel required. **(REQ ID: 12963BR).**

CONSULTANT(S) (DOMAIN) – US needed to help conduct IT requirements gathering, define problems, provide solution alternatives, create detailed computer system design documentation, implement deployment plan, and help conduct knowledge transfer with the objective of providing high-quality IT consulting solutions. **(REQ ID: 12964BR).**

CONSULTANT(S) (PRODUCTS AND PACKAGES) – US needed to help conduct IT requirements gathering, define problems, provide solution alternatives, create detailed computer system, design documentation, implement deployment plan, and help conduct knowledge transfer with the objective of providing high-quality IT consulting solutions. **(REQ ID: 12972BR).**

LEAD CONSULTANT(S) (DOMAIN) - US needed to anchor different phases of the IT engagement including business process consulting, problem definition, discovery, solution generation, design, development, deployment and validation. **(REQ ID: 12965BR).**

LEAD CONSULTANT(S) (PRODUCTS AND PACKAGES) - US needed to anchor different phases of the IT engagement including business process consulting, problem definition, discovery, solution generation, design, development, deployment and validation. **(REQ ID: 12971BR).**

PRINCIPAL(S) – Business Consulting needed to lead small proposals and multiple streams on complex proposals. Develop best in class proposals that present Infosys Point of View, approach and IT solution. Help identify clients and opportunities for the practice, present preliminary ideas and proposals to clients, lead engagements from launch to closure. Travel Required. **(REQ ID: 12970BR).**

PRINCIPAL CONSULTANT(S) (DOMAIN) - US needed to lead the engagement effort for IT assignments, from business process consulting and problem definition to solution design, development and deployment. Lead proposal development. Travel required. **(REQ ID: 12966BR).**

SENIOR TECHNOLOGY ARCHITECT(S) – US needed to provide IT architectural solutions for one or more projects. Provide input to create technology and architectural frameworks. Understand and analyze client business & IT problems, technology landscape, IT standards, and enterprise roadmaps. **(REQ ID: 12969BR).**

TECHNOLOGY ARCHITECT(S) - US needed to provide inputs on IT solution architecture based on evaluation/understanding of solution alternatives, frameworks and products. Will interact with clients to elicit architectural and non-functional requirements like performance, scalability, reliability, availability, maintainability. **(REQ ID: 12968BR).**

TECHNOLOGY ARCHITECT(S) (BUSINESS INTELLIGENCE) - US needed to provide inputs on IT solution architecture based on evaluation/understanding of solution alternatives, frameworks or products. Will interact with clients to elicit architectural and non-functional requirements like performance, scalability, reliability, availability, maintainability. **(REQ ID: 12967BR).**

15-NH-3482. **APPLICATIONS DEVELOPER _ in OVERLAND PARK, KS** to work with internal & external customers to translate business requirements into syst-level requirements & implement into a developed application. Job ID: 15-KS-2358. **APPLICATION SUPPORT ANALYST _ in OVERLAND PARK, KS** to ensure that the application is performing adequately & enough capacity is available to meet the business requirements & growth projections of the customer. Telecommuting is available from anywhere in the US. Job ID: 15-KS-2573. **ENGINEER – SERVICES SOFTWARE _ in PLANO, TX** to deliver services within site engineering, integration &/or configuration or support of products & ntwrks in accordance with a customer service contract. Frequent travel required. Job ID: 16-TX-2843. **ENGINEER – SERVICES RF _ in PLANO, TX** to perform radio network design, RF tuning & optimization for high capacity wireless networks using GSM, CDMA, LTE and WCDMA technologies. Job ID: 16-TX-1913. **PROJECT MANAGER _ in PLANO, TX** to establish project plan baseline by defining project scope, secure the necessary resources, & plan and monitor all activities. Job ID: 15-TX-2480. **BUSINESS CONSULTANT _ in PLANO, TX** to assist in the management and delivery of technology consulting projects. Job ID: 16-TX-2662. E**NGINEER – SOFTWARE _ in SANTA CLARA, CA** to architect, design & build end-to-end premium media service; shape the new look of television & entertainment experiences. Job ID: 16-CA-2373. **ENGINEER – SOFTWARE _ in WALTHAM, MA** to provide technical leadership with our MetraNet product User Interface (UI) & middle tier development. Job ID: 15-MA-3195. To apply, please mail resume to Ericsson Inc. 6300 Legacy Dr., R1-C12 Plano, TX 75024 & indicate appropriate job ID.

SPLUNK INC. has the following job opportunities in San Francisco, CA: Software **Engineer** (REQ#**9D735Q**). Utilize cloud infrastructure to provide high availability services to users using Splunk SaaS. **Software Engineer** (REQ#**9FC2G6**). Create robust, fault-tol distr sys in multi-threaded, multi-process environment. **IT Project Analyst** (REQ#**9QYSQA**). Facilitate IT portfolio mngmnt & dev of IT portfolio mngmnt office. Contribute to strategic, annual, and quarterly planning across mult lines of business. **Senior Release Engineer** (REQ#**9SSU8F**). Develop & support multi-platform build & release mngmnt tools & infrastructure to build, package & deploy co. prods. **Software Engineer**

(REQ#**9XBTQP**). Automate tests for sw features using Python-based test framework. **Software Engineer** (REQ#**9XQVSA**). Design, develop & deliver cloud based apps & sys for SaaS for users. **Senior Software Engineer** (REQ#**97R27H**). Develop & maintain the Co.'s core UI framework, APIs & tooling. **Software Engineer** (REQ#**9UBVFP**). Support Co.'s cloud eng team & app dev team to provide cloud infrastructure services to customers. **Senior Web Developer** (REQ#**9R8RSF**). Dev Content Mngmnt Sys apps, templates, & components based on packaged WCMS sw. **Senior Software Engineer** (REQ#**9RMTQ3**). Design, dev, test & sustain Co. apps. Authorize sw tests & provide sw solutions. **Splunk Inc.** has the following job opportunity in **Palo Alto, CA: Principal Software Engineer** (REQ#**9Z5VY9**). Architect, design & develop Co.'s ETL engine, data platform, connectors, parsers, & associated logic. **Splunk Inc.** has the following job opportunity in **Seattle, WA: Principal Product Manager** (REQ#**9UQQ32**). Define market position, determine user personas, define & execute UX R&D, create & prioritize feature back logs, & drive prod adoption. 25% travel req.

ENGINEERING

SanDisk Corporation

has openings in
Milpitas, California for:

TECHNICAL ROTATION ENGINEERS

To design, simulate, and verify analog IP (including regulators, oscillators, reference generators), of system on chip memory controllers. Job code: SD375.

To apply, reference job code # & mail resume to: SanDisk Corporation, 951 SanDisk Drive, MS: HRGM, Milpitas, CA 95035. EOE

Refer to Req# & mail resume to Splunk Inc., ATTN: J. Aldax, 250 Brannan Street, San Francisco CA 94107. Individuals seeking employment at Splunk are considered without regards to race, religion, color, national origin, ancestry, sex, gender, gender identity, gender expression, sexual orientation, marital status, age, physical or mental disability or medical condition (except where physical fitness is a valid occupational qualification), genetic information, veteran status, or any other consideration made unlawful by federal, state or local laws. To review US DOL's EEO is The Law notice please visit: https://careers.jobvite.com/Splunk/EEO_poster.pdf. To review Splunk's EEO Policy Statement please visit: http://careers.jobvite.com/Careers/Splunk/EEO-Policy-Statement.pdf. Pursuant to the San Francisco Fair Chance Ordinance, we will consider for employment qualified applicants with arrest and conviction records.

MARKETING ANALYST, Memphis, TN: Gather data, support, analyze, maintain marketing data/reports for IT resourcing; Work with vendors/end clients to gather data and prepare reports for management. Monitor marketing information/reports continuously for qualified candidates/vendors/end clients. Reply to: SVS Technologies Limited, 8700 Trail Lake Drive, #228 Memphis, TN 38125

ERICSSON INC. has openings for positions of: **SOLUTIONS ARCHITECT _ in BELLEVUE, WA** to support customer units & engagement practices; answer statements of compliance docs & develop solution descriptions. Job ID: 16-WA-3643. **BUSINESS CONSULTANT _ in BELLEVUE, WA** to drive key decisions & implementations; identify & translate customer's business scenarios & workloads into functional requirements. Requires 75% domestic travel. Job ID: 16-WA-2811. **SOLUTIONS ARCHITECT _ in BELLEVUE, WA** to provide customer requirements; work with account from pre-sales to implementation. Job ID: 16-WA-3501. **SOLUTIONS ARCHITECT _ in PLANO, TX** to launch & optimize telecom networks around the world & propose solutions based on customer

Intuit Inc.

has openings for the following positions in **Santa Clara County, including Mountain View, California** or any office within normal commuting distance:

Technical Data Analysts (Job code: I-2269): Understand the business and work to deliver business monitoring analytics. Collaborate with business stakeholders in acting on complex, multi-source data to explore, generate and test business assumptions. **Database Administrators (Job code: I-283):** Design and install database systems including setting up monitoring, backup and alert mechanism. **Senior Application Operations Engineers (Job code: I-482):** Develop highly scalable, secure and efficient software that support critical functions of Intuit's engineering operations and/or Intuit's leading commercial software products. **Product Managers (Job code: I-145):** Become an internal and external champion, evangelist, and expert on mobile payments for small businesses. Cross-functional development, marketing and support teams to execute on the GoPayment product strategy and roadmap. **Staff Software Engineers in Quality (Job code: I-453):** Partner with cross-functional leaders and team members to deliver Intuit products, with greater efficiency and speed.

Positions located in **Woodland Hills, California**:

Software Engineers in Quality (Job code: I-1376): Apply best software engineering practices to ensure quality of products and services by designing and implementing test strategies, test automation, and quality tools and processes. **Managers 3 Development (Job code: I-218):** Design and implement software applications, and discover and evaluate the most relevant factors to be considered in the creation, design, and implementation of software and services.

Positions located in **San Diego, California**:

Senior Software Engineers in Quality (Job code: I-489): Apply senior level software engineering practices and procedures to design, influence, and drive quality and testability of products and services. **Software Engineers (Job code: I-208)**: Apply software development practices to design, implement, and support individual software projects.

Positions located in **Plano, Texas**:

Senior Software Engineers (Job code: I-125): Exercise senior level knowledge in selecting methods and techniques to design, implement, modify and support a variety of software products and services to meet user or system specifications. **Software Engineers (Job code: I-945)**: Apply software development practices to design, implement, and support individual software projects.

To apply, submit resume to Intuit Inc., Attn: Olivia Sawyer, J203-6, 2800 E. Commerce Center Place, Tucson, AZ 85706.
You must include the job code on your resume/cover letter. Intuit supports workforce diversity.

requirements. Telecommuting is available for this position from anywhere in the US. Job ID: 16-TX-3697. **ENGINEER _ in PLANO, TX** to perform software loading, configuration, integration, verification, & troubleshooting of existing solutions on a customer site or in a lab environment using company products. Up to 20% domestic travel required. Job ID: 16-TX-3705. **BUSINESS CONSULTANT _ in PLANO, TX** to engage with clients to identify business needs, design solutions, and utilize idea packaging skills to propose defined frameworks and solutions. Up to 75% domestic travel required. Job ID: 15-TX-2098. **ENGINEER - SERVICES SOFTWARE _ in OVERLAND PARK, KS** to perform network analysis, software loading & configuration, verification, & optimization activities related to Ericsson's IMS networks, nodes, & related platforms. Requires 30% domestic travel. Job ID: 16-KS-1455. To apply, mail resume to Ericsson Inc. 6300 Legacy Dr., R1-C12, Plano, TX 75024 & indicate appropriate job ID.

OPTIMIZATION-based software company providing solutions for transportation scheduling and logistics problems has the following openings in Gainesville, FL: **Senior Manager (1)** – Gather, define and formalize business requirements and project objectives; prepare detailed task plan; supervise development and testing team. Master's degree (or Bachelor's degree, or foreign equivalent, and 5 years progressive experience in the field managing software development projects) in Operations Research, Computer Science, Industrial & Systems Engineering, or related field required. $100,000/year. Use ref. no. 20161. **Senior Systems Engineers (3)** - Conduct research on network and heuristic optimizations; lead algorithmic development team; develop prototypes. Ph.D. degree (or Master's degree and 2 years experience performing the job duties) in Operations Research, Computer Science, Industrial & Systems Engineering, or related field required. $90,000/year. Use ref. no. 20162. **Systems Engineer (1)** - Develop prototypes for optimization and simulation based software solutions. Master's degree (or Bachelor's degree, or foreign equivalent, and 5 years progressive experience performing the job duties) in Operations Research, Computer Science, Industrial & Systems Engineering, or related field required. $75,000/year. Use ref. no. 20163. **Software Engineer (1)** – Design, implement and test optimization and simulation software solutions. Master's degree (or Bachelor's degree, or foreign equivalent, and 5 years progressive experience performing the job duties) in Computer Science, Operations Research, Industrial & Systems Engineering, or related field required. $80,000/year. Use ref. no. 20164. Full-time. Send resume by mail, specifying the specific position, with reference number, you are applying for to: HR, Innovative Scheduling, LLC., 7600 NW 5th Place, Gainesville, FL 32607, or, by email to job.us@optym.com.

PROGRAMMER ANALYST- design, develop, test & implement applications utilizing knowledge of.Net, SharePoint Portal technologies like Windows SharePoint Server(WSS 3.0), Microsoft SharePoint (SP2010) and SharePoint 2013, SQL Server, Microsoft Info-path forms 2010 /2013, Nintex Forms and Workflows. Must be willing to travel & reloc to unanticipated client locations throughout the US. Reqs MS in comp sci, eng or rel. Mail resumes to Strategic Resources International, Inc. 777 Washington Rd, Suite 2, Parlin, NJ 08859

4XVENTURES LLC seeks Dist'd. Engg (Strategy, Design, Architect) & VP Engg. (Lead, Plan, Manage, Architect) in Menlo Park, CA. Must have Bachelor's deg./equi. in CS, CA, CIS, MIS, Engg and 5+ years in rel. field. Travel reqd. Apply: Careers@4xventures.com

MANAGER. Job location: Miami, FL & any other unanticipated locations in U.S. Travel Required. Duties: Identify & resolve finan. systems issues critical to clients' strategic & oper. success. Gather business & design requirements among stakeholders enterprise wide & develop detailed design doc. to align with client's requirement & specs. Provide tech/func. content. Develop & present conclusions & recommends. to sr. client mgmt. Develop detailed admin. guide for end users using Oracle & SAP. Lead design of plan, budget & forecast processes. Manage client relationships & report project progress. Requires: M.S. degree in Comp. Sci., Acctg. or related field & 3 yrs. exp. in the job offered or 3 yrs. exp. as a Consultant or Audit Ass't. Will accept B.S. (or foreign equiv.) & 5 yrs. exp. in finance/acctg. ind. in lieu of M.S. & 3 yrs. exp. Concurrent exp. must incl.: 3 yrs. exp. with finan. systems.; 3 yrs. exp. leading design of plan, budget & forecast processes; & 3 yrs. exp. with SAP. Send resume (no calls) to: Michelle Ramirez, The Hackett Group, Inc., 1001 Brickell Bay Dr., Suite 3000, Miami, FL 33131.

CLASSIFIED LINE AD SUBMISSION DETAILS: Rates are $425.00 per column inch ($640 minimum). Eight lines per column inch and average five typeset words per line. Send copy at least one month prior to publication date to: Debbie Sims, Classified Advertising, *Computer* Magazine, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720; (714) 816-2138; fax (714) 821-4010. Email: dsims@computer.org.

In order to conform to the Age Discrimination in Employment Act and to discourage age discrimination, *Computer* may reject any advertisement containing any of these phrases or similar ones: "...recent college grads...," "...1–4 years maximum experience...," "...up to 5 years experience," or "...10 years maximum experience." *Computer* reserves the right to append to any advertisement without specific notice to the advertiser. Experience ranges are suggested minimum requirements, not maximums. *Computer* assumes that since advertisers have been notified of this policy in advance, they agree that any experience requirements, whether stated as ranges or otherwise, will be construed by the reader as minimum requirements only. *Computer* encourages employers to offer salaries that are competitive, but occasionally a salary may be offered that is significantly below currently acceptable levels. In such cases the reader may wish to inquire of the employer whether extenuating circumstances apply.

LinkedIn Corp.

LinkedIn Corp. has openings in our **Mtn View, CA** location for:

Software Engineer (All Levels/Types) (SWE116MV) Design, develop & integrate cutting-edge software technologies; **Engineering Manager (6597.1549**) Lead a team focused on data lifecycle management across Hadoop clusters; **Manager, Software Engineering (6597.121)** Lead a team of Software Engineers to help scale LinkedIn's infrastructure to handle massive growth in membership, traffic, & data; **Web Developer (6597.919)** Own front-end development for products & collaborate with visual/interaction designers, engineers, & product managers to launch new products, iterate on existing features, & build a world-class user experience; **Technical Program Manager (6597.428)** Design, implement & maintain software engineering project management process; **Test Engineer (6597.1266)** Drive test development of services and distributed systems.

LinkedIn Corp. has openings in our **Sunnyvale, CA** location for:

Software Engineer (All Levels/Types) (SWE116SV) Design, develop & integrate cutting-edge software technologies; **Test Engineer (6597.1157)** Design, code, implement, test, analyze & deploy application or test software; **Senior Product Manager (6597.1259)** Understand & lead analyses of the competitive environment, customers & product metrics to determine the right set of features to drive engagement & usage; **Technical Solutions Architect, Sales Systems (6597.1191)** Responsible for the analysis, design, & development of sales systems projects in conjunction with other members of Sales Systems teams. Limited travel required to other company worksites in the Bay Area.

LinkedIn Corp. has openings in our **San Francisco, CA** location for:

Software Engineer (All Levels/Types) (SWE116SF) Design, develop & integrate cutting-edge software technologies.

LinkedIn Corp. has openings in our **New York, NY** location for:

Senior Test Engineer (6597.32) Design & develop advanced test suites using object-oriented methodologies.

LinkedIn Corp. has openings in our **Calabasas, CA** location for:

Senior Quality Assurance Engineer (6597.1349) Work with agile team members in estimating test effort using SCRUM & Kanban methodologies.

Please email resume to: 6597@linkedin.com. Must ref. job code above when applying.

Apple Inc. has the following job opportunities in Cupertino, CA:

ENGINEERING

Software Systems Engineer (Req#9K-9TLL) Validate media frameworks by creating white-box test tools that verify functionality, prfrmnc, & stability in the area of media playback, editing, export, & capture.

User Experience Design Engineer (Req #9D2TWY). Define user exp reqs for feats of Apple consumer apps.

Product Design Engineer (Req#9T-P4RN) Dvlp glazing syss (glass/plastic covers) for optimum sound and vibration prfmnce.

Software Development Engineer (Req#9H3T7V) Dsgn, dvlp, & integrate high prfmnce image processing SW for embedded syss.

Software Quality Assurance Engineer (Req# 9MC2EL). Dev, des, & execute tests for valid of gas gauge HW & FW at the sys lvl, dev test plans & other test document as req'd.

Software Systems Engineer (Req# 9C5MF6). Dev SW & FW using C/C++ or Objective C programming lang. Travel req'd 25%.

Software Quality Assurance Engineer (Req#9UYNX5). Dev., design, and execute manual and automated tests for compatibility of Apple HW and SW projects

Software Engineer Systems (Req#9G83BU). Define rendering specs. for current and future products. Implement camp. vision algorithms.

Software Engineer Applications (Req#9XXQ7W). Dev. & maintain Java based NoSQL dbase infrastruct & apps through automation

Software Engineer Applications (Req#9ZX45D). Build high perf., highly scalable, fault tolerant backends for critical internal sys.

Software Development Engineer (Req#9V6UZG). Design, dev., and support SW sys. for info. retrieval and external facing search experience. Travel req. 15%.

Software Development Engineer (Req# 9FSV8E). Analyze comp HW and SW problems, determine the cause, assess the prob's risk and priority levels, and determine app. solutions for internal eng. teams.

ASIC Design Engineer (Req#9D-MQAB). Architect, dev., maintain and enhance simulation flows and methodology for analog/mixed-signal (AMS) and RF circuit designs

Software Development Engineer (Req#9D327A). Design, dev. and execute SW valid suites for complex networking protocols used in the commun of Apple devices over the netwrk.

Hardware Development Engineer (Req#9NTP26). Dsgn. and dev. capacitive sensor elements for Apple products. Travel req. 25%

User Experience Designer (Req#9EYU7Y) Prpose interactive solutions & doc w/ site maps, user flows, & wireframes.

Hardware Development Engineer (Req#9D6VLC) Res for program mgmt of NAND flash memory component in Apple sys.

User Experience Designer (Req#-9JUQL2) Prpose interactive solutions & doc w/site maps, user flows, & wireframes.

Host and Web Security Engineer (Req#9UEVFA) Progm & sup in-house des'd & maint'd web app firewall & SW load-balancer solution.

Firmware Engineer (Req#9LUS6U) Rsrch, dsgn, dvlp, implmnt, & debug wireless audio firmware.

Software Engineer Systems (Req#9X-QM7Y). Research, des, dev, and impl robust real-time sys to perceive the environment

Software Development Engineer (Req#9PQ39S) Dvlp, doc, & execute test plans & procedures in coordination with the team.

Software Engineer Applications (Req# 9EP3G2). Des, bld & supp new critical infrastructural sys & FW's.

Software Development Engineer (Req#9EYTU2) Des & dev image process algo for camera features.

Hardware Development Engineer (Req#9E5QPX) Spprt strctral, thrml, fluid, & dynmc finite elmnt anlyss for sensing & dsply technlgies.

ASIC Design Engineer (Req#9FN372) Verify high prfmnce microprocessor dsgns.

Software Engineer Applications (Req#9KDME8) Dsgn & dvlp Apple cloud srvces, spprtng 100s of millions of customers.

Product Design Engineer (Req#9F-NP9N) Conceive, dsgn, & prdce new pwr adptrs for prdcts. Travel req. 20%.

Software Development Engineer (Req # 9KHPNV). Test iOS telephony func on Apple iOS devs from cell perspctve view.

ASIC Design Engineer (Req#9JHRQF) Extrct & validate the dsgn parameters from phys dsgn such as PCB & package.

Product Design Engineer (Req#9W3VBK) Des new input device prods, subsys, & mods to sup Mac & iOS prod lines.

Software Development Engineer (Req#9F4VVF) Archtct, dsgn, & dvlp iOS Wallet app & associated SW components.

Systems Design Engineer (Req#9RUUMY). Des and dev internal SW tools and sys for the Wireless Des dept at Apple.

Engineering Program Specialist (Req #9FZ28U). Coord & plan materials for hw eng progs w/ exposure to cross-func teams & overseas vends.

Software Development Engineer (Req # 9TTUYY). Dsgn Maps Data Insight Gathering & Validation Framewk comp using Big Data techs.

ASIC Design Engineer (Req # 9GNUWH). Dsgn & dev multimedia IP's & subsys in SoC ASICs.

Software Development Engineer (Req#9GANBJ) Triage, route, and resolve GPU driver defects. Dev GPU Driver SW, including the developer APIs (Metal, OpenGL, and OpenCL) to the low lvl kernel-mode driver infrastruc.

Software Engineer Applications (Req#9JPUG8) Dsgn & dev middleware svc layer components for EDM, EDW, & BigData reporting platform

ASIC Design Engineer (Req#9G-C4AA) Create SW to verify architecture & func of pre-silicon HW designs.

Hardware Development Engineer (Req#9H8QSX) Dsgn transducers, select materials, & prfrm characterization & simulation.

Engineering Project/Program Manager (Req#9ML2Y6) Plan & exec Apple Power Eng Prog through bldg a strat vision, impl proc & providing precise guidance to the team while leveraging eng & op partners to meet & exceed the prog critical milestones. Travel req'd 25%.

Software Engineer Applications (Req#9XYR36) Dsgn, dvlp, certify & support paymnt SW for Apple's Point of Sale System.

Software Development Engineer (Req #9U8TY3) Conduct Bluetooth testing of iOS devices. Travel req'd 15%.

ASIC Design Engineer (Req # A2M2SG) Design sys level test automation prog & HW for debug, charac, qualification, & prod of SoC devices. Travel req'd 20%.

IST Technical Project Lead (Req# 9UCPV8). Prfrm & coordinate cmpliance audit & assessment testing as needed to meet reqs of multiple compliance orgs.

Software Engineer in Testing (Req#9N8UQR) Sup delivery of high qual apps for iOS & OS X platforms.

Info Systems Engineer (Req#9T73UP) Des & dev scalable, high perfrm'g portal solutions.

ASIC Design Engineer (Req#9JUQWS) Des digital blocks in mixed signal circts like Analog-to-Digital circuits (ADC) and Digital-to-Analog circuits (DAC).

Software Engineer Applications (Req#A2P2Z9). Build high perf, highly scalable, fault tolerant backends for critical internal sys

Software Development Engineer (Req #9TPS6W) Dev & maintain low-level app prog interfaces which provide oper sys support for multithreading & inter-process comm.

Software Development Engineer (Req#9VP2EH). Des and dev SW for iBooks with specific emphasis on close collab with WebKit team and code base.

Software Engineer Applications (Req#9V7VW3). Translate user req's & sys. objectives into work-flow designs, programs in a data whouse & reporting envir

Software Engineer Applications (Req#9W2N4V). Des, dev, & maintain highly-perf internet scale web services that handle large vol of concur transactions.

Software Development Engineer (Req#9E9TAT). Create test plans for new and existing first-party apps and OS functionality

Software Development Engineer (Req#9FFRJP) Des & dev SW for search engines on comps & personal comm devices.

Technical Product Lead (Req#9ZT38B). Build a platform for iTunes business using latest and best in industry tech and solutions.

Human Fact Design Engineer (Req#9DP2EL) Rspnsbl for the dsgn & dvlpmnt of user interface for Appl's Consumer SW Apps for Mac & iOS operating sys.

Software Engineer Applications (Req #9QF2CH). Dev, build, & manage highly scalable infrastructure for hosting internet prod including iTunes &Maps at Apple data centers.

Software Engineer Applications (Req#9YDUVE). Build and maintain highly scalable, distributed Enterprise platforms

Engineering Project Coordinator/ Specialist (Req#9WEQAL) Resp for sourcing, vendor manufact dev execution & supply ramp to mass production for Apple req raw materials. Travel Req 35%.

Web Developer (Req#9WG2LD) Dev self-serv'd apps, integrate busi crit sys, & innovate new solutions w/ exst'g tools & tech.

Product Design Engineer (Req#9PPV5S) Des eng'ng proc's for adv manufact'g & PCB tech's & plan projs for new tech dev'ment.

Hardware Development Engineer (Req#9F4R3G). Des & sim electro-acous spkr and recvr mods used in iPhone, iPad, Mac, Watch and accssrs. Travel req'd: 20%.

Apple Inc. has the following job opportunities in Seattle, WA:

Software Engineer (Req#9Z526V) Dsgn, implmnt, & tune algorithms for the detection & correction of flaws in base map data.

Software Engineer Applications (Req #9VDRYP) Design & dev distributed SW to power iCloud apps & svcs.

Apple Inc. has the following job opportunity in Austin, TX:

ASIC Design Engineer (Req #9E2Q6L). Implement complex SOC blocks from netlist to GDS for state-of-the-art SOCs in latest tech nodes.

Refer to Req# & mail resume to Apple Inc., ATTN: L.J., 1 Infinite Loop 104-1GM, Cupertino, CA 95014. Apple is an EOE/AA m/f/ disability/vets.

Apple Inc. has the following job opportunity in Cupertino, CA:

Industrial Designer (Req#9JQTNL) Dvlp high quality dsgn concepts to drive industrial dsgn for new Apple prdcts. Travel req. 15%.

Interested applicants must submit a portfolio that demonstrates skills required. Please enclose a self-addressed stamped envelope if you wish your portfolio to be returned. Refer to Req# & mail resume to Apple Inc., ATTN: L.J., 1 Infinite Loop 104-1GM, Cupertino, CA 95014. Apple is an EOE/AA m/f/ disability/vets.

It's work that matters. It's what we do at Symantec. Symantec is the world leader in providing solutions to help individuals and enterprises assure the security, availability, and integrity of their information. In essence, we protect the free flow of information in a connected world. As the fourth largest independent software company in the world, Symantec has operations in more than 40 countries with 475 out of Fortune's global 500 companies using our solutions. People look to us to safeguard the integrity of their information, ensuring it is secure and available. Achieving this ambitious goal is only possible through the combined efforts of the innovators and visionaries that Symantec continuously attracts. Symantec draws the very best people with a variety of backgrounds, experiences and perspectives and provides them with a work environment where uniqueness is valued and empowered. The creative people we attract help define the spirit of innovation at Symantec. Symantec is proud to be an equal opportunity employer. We currently have openings for the following positions (various levels/types):

Addison, Texas

Sr. Technical Education Staff Members (1648.2063) Develop enablement materials for Symantec's Data Loss Prevention products, including Network Monitor, Network Prevent for Email, Network Prevent for Web, Network Discover, Network Protect, Data Insight, Cloud Storage, Cloud Prevent for Microsoft Office 365, Endpoint Prevent and Endpoint Discover. Must be available to work on projects at various, unanticipated sites throughout the US. May Telecommute.

Culver City, California

Engineering Managers (EMCC116) Direct and supervise team of engineering (QA and/or development teams). Develop standards for products and/or oversee development and execution of software and/or analysis of test results. Plan, design, develop and implement processes. **Product Managers (PDMCC116)** Participate in all software product development life cycle activities. Move software products through the product development cycle from design and development to implementation and testing. **Program Managers (PMCC116)** Work closely with engineering members, managers, and leads, product managers, ensure rapid execution and on time, high quality delivery of complex Data Loss Prevention (DLP) projects. **Software Engineers (SWECC116)** Responsible for analyzing, designing, debugging and/or modifying software; or evaluating, developing, modifying, and coding software programs to support programming needs. **Software QA Engineers (SQACC116)** Responsible for developing, applying and maintaining quality standards for company products. Develop and execute software test plans. Analyze and write test standards and procedures. **Threat Analyst Engineers (TAECC116)** Develops and maintains threat analysis content within the constraints of the short time-frame required to deliver this content to customers. Conceptual understanding and applied experience with security concepts and networking.

Herndon, Virginia

Information Security Analysts (Sr. MSS Services Manager) (SECVA116) Responsible for leading a team of service delivery managers, who represent the voice of the customer into the Symantec MSS organization. Drive the resolution of issues that are not getting resolved through normal incident and problem management processes. Some travel required. Must be able to travel 10% of the time with short notice. **Information Security Analysts (ISAVA116)** Responsible for leading a team of service delivery managers, who represent the voice of the customer into the Symantec Managed Security Services organization. Drive the resolution of issues that are not getting resolved through normal incident and problem management processes. **Operations Managers (OPSVA116)** Responsible for providing the highest level of support to internal and external customers as well as owning major functions within the Security Engineering organization. Provide formal supervision of Security Engineers and Sr. Security Engineers, including day-to-day management of the overall work queue including capacity planning, and recruitment.

Submit resume to JOBADS@symantec.com. Must reference position & code listed above. EOE.
For additional information about Symantec and other positions visit our website at http://www.symantec.com.

It's work that matters. It's what we do at Symantec. Symantec is the world leader in providing solutions to help individuals and enterprises assure the security, availability, and integrity of their information. In essence, we protect the free flow of information in a connected world. As the fourth largest independent software company in the world, Symantec has operations in more than 40 countries with 475 out of Fortune's global 500 companies using our solutions. People look to us to safeguard the integrity of their information, ensuring it is secure and available. Achieving this ambitious goal is only possible through the combined efforts of the innovators and visionaries that Symantec continuously attracts. Symantec draws the very best people with a variety of backgrounds, experiences and perspectives and provides them with a work environment where uniqueness is valued and empowered. The creative people we attract help define the spirit of innovation at Symantec. Symantec is proud to be an equal opportunity employer. We currently have openings for the following positions (various levels/types):

Mountain View, California

Business Systems Analysts (BSAHQ116) Analyze complex business problems to be solved with automated systems. Provide technical expertise in identifying, evaluating and developing systems and procedures that are cost effective and meet user requirements. **Business Intelligence Analysts (BIAHQ116)** Responsible for designing, developing and maintaining high-quality code for simple to complex anti-spam effectiveness tools, metrics or applications. **Computer Systems Analysts (CSAHQ116)** Analyze engineering, business and/or other business intelligence issues for application to Symantec solutions; and provide operational support in the development and implementation process of computer software applications, systems or services. **Network Systems Engineers (NSEHQ116)** Design, architect and maintain network system. Engage in capacity planning for application and performance management of the network. **Product Managers (PDMHQ116)** Participate in all software product development life cycle activities. Move software products through the product development cycle from design and development to implementation and testing. **Product Managers (1648.916)** Develop company market requirements for technical products or product lines, including product strategy definition, requirements analysis, and/or pricing. **Program Managers (PMHQ116)** Work closely with engineering members, managers, and leads, product managers, ensure rapid execution and on time, high quality delivery of complex Data Loss Prevention (DLP) projects. **Software Engineers (SWEHQ116)** Responsible for analyzing, designing, debugging and/or modifying software; or evaluating, developing, modifying, and coding software programs to support programming needs. **Software QA Engineers (SQAHQ116)** Responsible for developing, applying and maintaining quality standards for company products. Develop and execute software test plans. Analyze and write test standards and procedures. **UI Designers (UIHQ116)** Responsible for providing User Interface (UI) design and support to product development teams, including the design, analysis and investigation of applications, systems, and Graphic User Interfaces (GUIs).

San Francisco, California

Software Engineers (SWESF116) Responsible for analyzing, designing, debugging and/or modifying software; or evaluating, developing, modifying, and coding software programs to support programming needs. **Software QA Engineers (SQASF116)** Responsible for developing, applying and maintaining quality standards for company products. Develop and execute software test plans. Analyze and write test standards and procedures.

Springfield, Oregon

Operations Research Analysts (RAOR116) Operating in an intra/inter-function business and data analysis environment with emphasis on project work, responsible for evaluating, developing and implementing operations processes, procedures, programs and strategies to increase technical and operational efficiencies both within the group and other functions and to deliver consistent planning and controls across multiple locales.

Submit resume to JOBADS@symantec.com. Must reference position & code listed above. EOE.
For additional information about Symantec and other positions visit our website at http://www.symantec.com.

CALL FOR NOMINEES

Education Awards Nominations

Taylor L. Booth Education Award

A bronze medal and US$5,000 honorarium are awarded for an outstanding record in computer science and engineering education. The individual must meet two or more of the following criteria in the computer science and engineering field:

- Achieving recognition as a teacher of renown.
- Writing an influential text.
- Leading, inspiring or providing significant education content during the creation of a curriculum in the field.
- Inspiring others to a career in computer science and engineering education.

Two endorsements are required for an award nomination.

See the award information at:
www.computer.org/web/awards/booth

Computer Science and Engineering Undergraduate Teaching Award

A plaque, certificate and a stipend of US$2,000 is awarded to recognize outstanding contributions to undergraduate education through both teaching and service and for helping to maintain interest, increase the visibility of the society, and making a statement about the importance with which we view undergraduate education.

The award nomination requires a **minimum of three endorsements**.

See the award details at:
www.computer.org/web/awards/cse-undergrad-teaching

Deadline: 15 October 2016
Nomination Site: awards.computer.org

IEEE computer society

Move Your Career Forward

IEEE Computer Society Membership

Explore These Security Resources

Advance Your Career

Security Certificate of Achievement

The Computer Society now offers a Certificate of Achievement in the growing field of security. Take advantage of this professional development opportunity to increase your expertise and advance your career by successfully completing these four courses:

- Foundations of Software Security
- Secure Software Design
- Managing Secure Software Development
- Secure Software Coding

Skillsoft Featured Topic – Security

Explore relevant, up-to-date cybersecurity resources in Skillport. Use the resources to help solve a problem you're currently facing or for long-term training and professional education to help advance your career.

Build Your Knowledge

IEEE CYBER SECURITY

IEEE Cybersecurity Initiative

Launched in 2014, the initiative aims to improve the understanding of cybersecurity by students, educators, and professionals. Currently, the initiative is tackling computer security education, a building code for security critical software, under-represented groups in security, and many other aspects of this crucial subject. It has already launched the IEEE Center for Secure Design (CSD), which seeks to shift the focus in security from finding bugs to identifying common design flaws and building security in from the onset.

FOR DIRECT LINKS TO THESE RESOURCES, VISIT

www.computer.org/edge-resources

IEEE computer society

CELEBRATING 70 YEARS

IEEE computer society

ROCK STARS OF BIG DATA

Experience the Newest and Most Advanced Thinking in Big Data Analytics

24 May 2016 | Austin, TX

Big Data: Big Hype or Big Imperative?

BOTH.

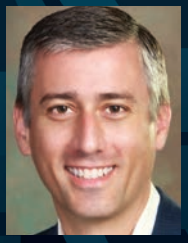
Business departments know the promise of big data—and they want it! But newly minted data scientists can't yet meet expectations, and technologies remain immature. Yes, big data is transforming the way we do—everything. But knowing that doesn't help you decide what steps to take tomorrow to assure your company's future.

That's why May 24 is your real-world answer. Come meet the experts who are grappling with and solving the problems you face in mining the value of big data. You literally can't afford to miss the all new Rock Stars of Big Data 2016.

Rock Star Speakers

Kirk Borne
Principal Data Scientist, Booz Allen Hamilton

Satyam Priyadarshy
Chief Data Scientist, Halliburton



Bill Franks
Chief Analytics Officer, Teradata

www.computer.org/bda