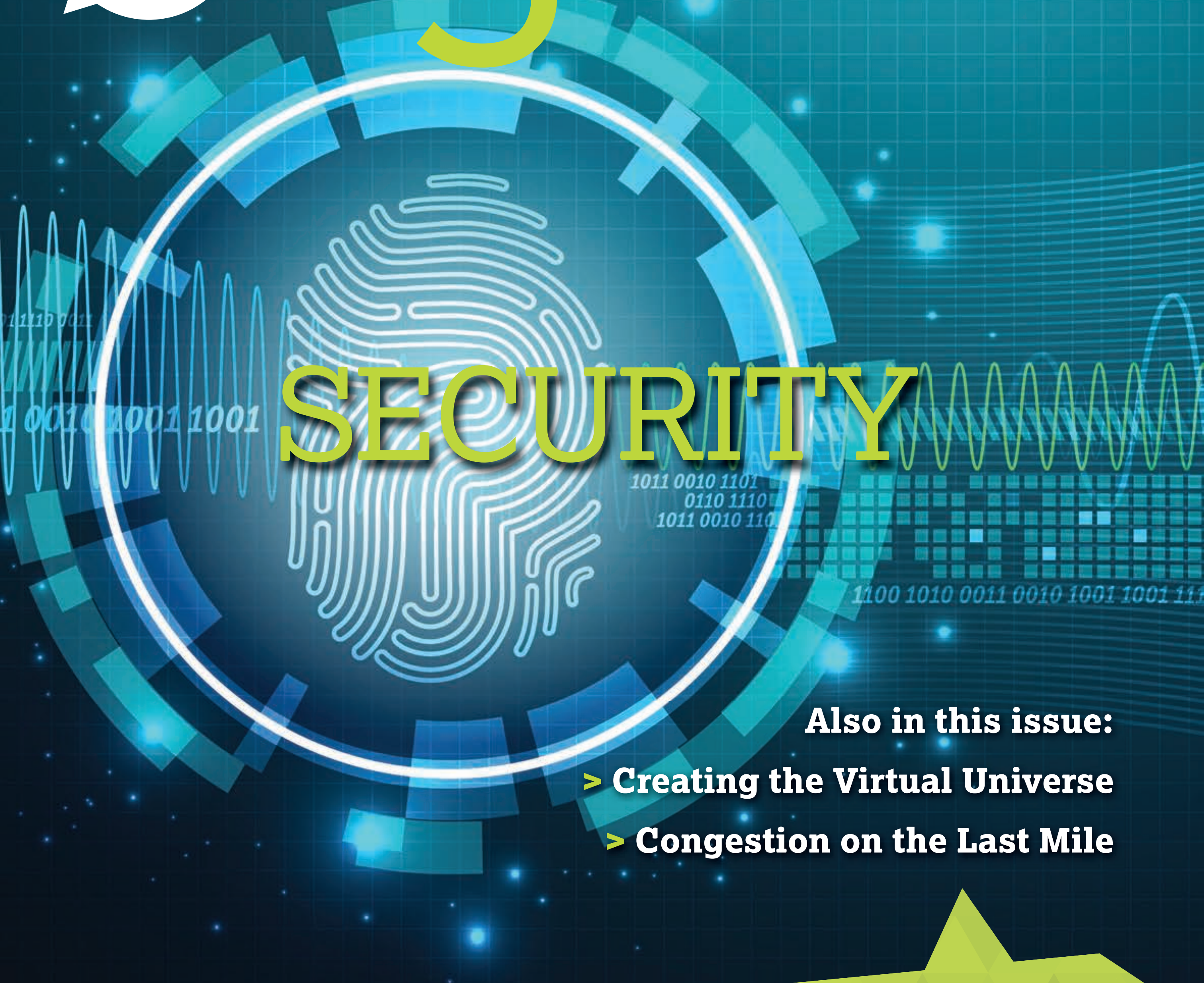
COMPUTING edge



SECURITY

Also in this issue:

> Creating the Virtual Universe

> Congestion on the Last Mile

FEBRUARY 2017

www.computer.org

IEEE

IEEE computer society

IEEE computer society
Now there's even more to love about your membership...

NO
ADDITIONAL
FEE
myCS
Computer
BIG DATA
Computer

COMPUTING edge

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

Printed with inks containing soy and/or vegetable oils

IEEE

IEEE computer society

STAFF

Editor
Lee Garber

Contributing Staff
Christine Anthony, Brian Brannon, Lori Cameron, Cathy Martin, Chris Nelson, Meghan O'Dell, Dennis Taylor, Rebecca Torres, Bonnie Wylie

Production & Design
Carmen Flores-Garvey, Monette Velasco, Jennie Zhu-Mai, Mark Bartosik

Acting Editorial Content Manager
Carrie Clark

Senior Manager, Editorial Services
Robin Baldwin

Director, Products and Services
Evan Butterfield

Senior Advertising Coordinator
Debbie Sims

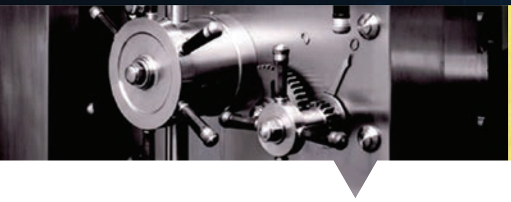
Circulation: ComputingEdge (ISSN 2469-7087) is published monthly by the IEEE Computer Society. IEEE Headquarters, Three Park Avenue, 17th Floor, New York, NY 10016-5997; IEEE Computer Society Publications Office, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720; voice +1 714 821 8380; fax +1 714 821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

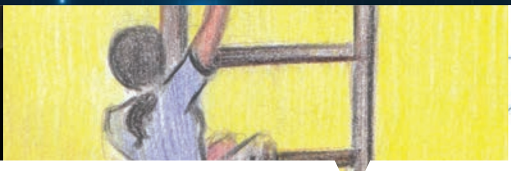
Postmaster: Send address changes to ComputingEdge-IEEE Membership Processing Dept., 445 Hoes Lane, Piscataway, NJ 08855. Periodicals Postage Paid at New York, New York, and at additional mailing offices. Printed in USA.

Editorial: Unless otherwise stated, bylined articles, as well as product and service descriptions, reflect the author's or firm's opinion. Inclusion in ComputingEdge does not necessarily constitute endorsement by the IEEE or the Computer Society. All submissions are subject to editing for style, clarity, and space.

FEBRUARY 2017 • VOLUME 3, NUMBER 2

COMPUTING edge





18

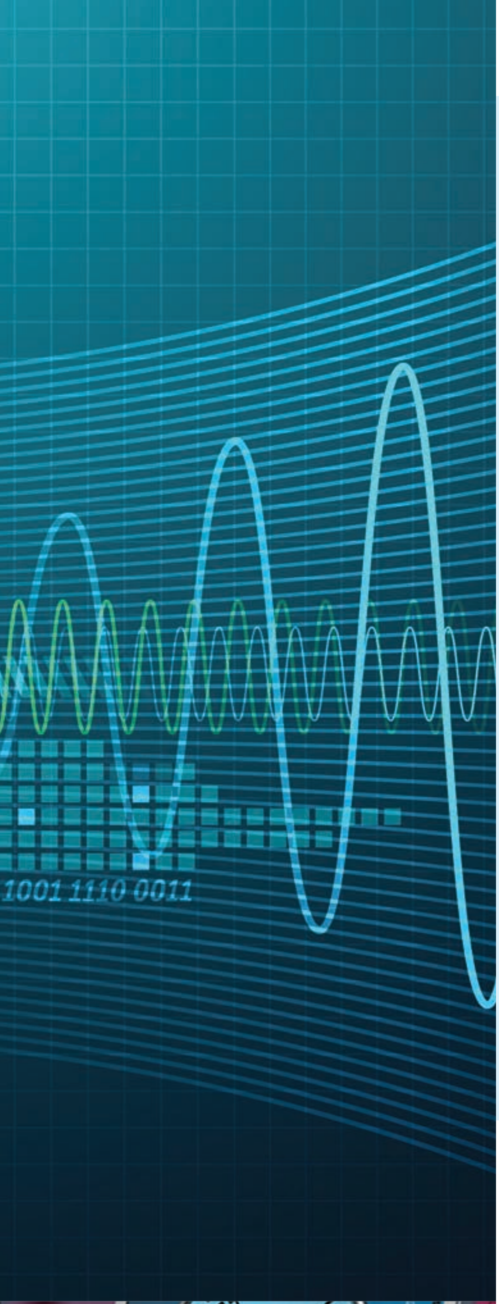
Defeating Buffer Overflow: A Trivial but Dangerous Bug

22

The Outcomes of Cybersecurity Competitions and Implications for Underrepresented Populations

28

Cloud Manufacturing: Security, Privacy, and Forensic Concerns



35

Evil Offspring–Ransomware and Crypto Technology

Departments

CS FOCUS

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

Human-augmentation technologies can help users enhance existing abilities and give them some they lack. These technologies are the focus of *Computer*'s February 2017 special issue.

IEEE Software

No consolidated set of software engineering best practices for the Internet of Things (IoT) has yet emerged. Too often, unprepared programmers put together IoT systems in an ad hoc fashion and release them into the marketplace, often poorly tested. *IEEE Software*'s January/February 2017 special issue aims to provide the basis for a set of best practices that will guide the industry through the challenges of **software engineering for the IoT**.

IEEE Internet Computing

Network function virtualization (NFV) represents a series of technologies that let users virtualize the high-volume packet-processing functions that form the Internet's core so that the functions can run on commodity cloud-computing platforms. NFV will spur innovation and enable the faster deployment of new services with less risk, according to *IEEE Internet Computing*'s November/December 2016 special issue on the topic.

Computing in Science & Engineering

As we enter the Internet of Things era, in which lightweight mobile devices become the main online terminals, **transparent computing** provides opportunities and presents challenges. *CiSE*'s January/February 2017 special issue highlights this new paradigm.

IEEE Security & Privacy

Considerable work is taking place on the interface between cryptography practice and theory. The articles in *IEEE S&P*'s November/December 2016 special issue show that **real-world cryptography** no longer focuses

Published by the IEEE Computer Society 2469-7087/17/$33.00 © 2017 IEEE

only on the traditional aspects of communications security. The articles also demonstrate that practitioners are concerned about cryptography's societal impacts and underlying social constructs.

IEEE Cloud Computing

IEEE Cloud Computing's November/December 2016 special issue addresses the use of **cloud computing for enhancing living environments**.

IEEE Computer Graphics and Applications

CG&A's January/February 2017 special issue on **water, sky, and the human element** includes articles on a natural interface for underwater robots' remote operation, a decision-support application for a sustainable water-distribution system, the real-time visual tracking of deformable objects in robot-assisted surgery, and a machine-learning-driven sky-illumination model.

IEEE Intelligent Systems

"On Searching the Internet of Things: Requirements and Challenges," from *IEEE Intelligent Systems*' November/December 2016 issue, describes some of the requirements of and key challenges to building scalable and efficient **search and discovery mechanisms for the Internet of Things**.

IEEE MultiMedia

According to the authors of "A Neural Network for Quality of Experience Estimation in Mobile Communications," from *IEEE MultiMedia*'s October–December 2016 issue, we need a new way to express multimedia-service users' satisfaction: **quality of experience** (QoE). They consider key performance indicators (KPIs) and propose using neural networks to automatically classify these KPIs in terms of QoE.

IEEE Annals of the History of Computing

The authors of "The Dawn of Digital Light," from *IEEE Annals*' October–December 2016 issue, say the first digital images—still photos, videogames, and computer animations—were made on early computers in the late 1940s and early 1950s. This fresh perspective on digital pictures establishes a different take on the history of early computers and unifies **the history of digital images**.

IEEE Pervasive Computing

Initial **drone research** was mostly concerned with improving technical capabilities, such as battery life and flight accuracy. More recent research investigates how drones can support existing application domains and even create new ones. *IEEE Pervasive Computing*'s January–March 2017 special issue discusses this more recent work. In addition, instead of looking at the type of large drones used by the military, the issue focuses on smaller drones that fly at lower altitudes, which could play a more significant role in pervasive applications.

IT Professional

Information and communications technology (ICT) environments have dramatically changed in recent yoears. They now include complex distributed architectures and mission-critical services and applications. However, determining whether these services and applications are correctly coded against attacks and other problems can be difficult. In "**Practical Correctness in ICT Environments**," from *IT Pro*'s November/December 2016 issue, the author examines this concern and presents possible solutions.

IEEE Micro

To reach its potential, the Internet of Things (IoT) must break down the silos that limit applications' interoperability and hinder their manageability. Doing so would enable the building of ultra-large-scale systems (ULSSs). To deal with the resulting complexity, the authors of "Emergent Behaviors in the Internet of Things: The Ultimate Ultra-Large-Scale System," from *IEEE Micro*'s November/December 2016 issue, propose **hierarchical emergent behaviors** (HEB).

Computing Now

The Computing Now website (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.

UPCOMING EVENTS

Register for the Computer Society's Exclusive TechIgnite Event

The IEEE Computer Society's TechIgnite event is now open for registration. This exclusive two-day event takes place 21–22 March 2017 at the Hyatt Regency in Burlingame, California. The event, whose theme is "The Truth behind Technology," is designed to empower tech professionals in the areas of cybersecurity, blockchain, machine learning, quantum computing, operational intelligence, 5G wireless, and virtual reality. Attendees will hear from 33 tech gurus about the real and perceived dangers and benefits associated with emergent trends, including artificial intelligence, deep learning, augmented reality, and more.

The event includes two fireside chats with world-renowned technology leaders Steve Wozniak, cofounder of Apple Computer; and Grady Booch, IBM chief scientist of software engineering. Also speaking will be US Department of Homeland Security chief technology officer Peter Fonash, GE Digital chief executive officer Bill Ruh, renowned futurist Brian David Johnson, and Medtronic vice president Annette Brüls. In addition, more than 20 other widely recognized IT leaders will appear.

Attendees can engage in lively panel discussions addressing key industry challenges, explore more than 40 exhibits, and network with thousands of other IT professionals from many industries.

For a complete lineup of speakers and to register, visit www.computer.org/techignite.

2469-7087/17/$33.00 © 2017 IEEE

EDITOR'S NOTE

Making the World of Computing More Secure

Security is a concern in almost every area of computing. We want our software, hardware, and networks to be secure. We want to conduct business on the Internet without fretting about identity theft, we want to communicate with others without concern about eavesdropping, and we want to open emails without worrying about downloading malware.

This *ComputingEdge* issue explores these concerns and looks at some of today's most important security-related matters.

A "cyberfog" security approach that splits data into numerous fragments and disperses them across multiple devices could provide attack resiliency but also presents formidable technical challenges, according to *Computer*'s "The Fog of War in Cyberspace."

In *IEEE Security & Privacy*'s "Stop Trying to Fix the User," author Bruce Schneier contends that the problem with security isn't users but the poor design of system security that forces them to do counterintuitive things.

Intelligent interfaces can provide high-quality, contextually relevant user experiences. However, they also raise privacy concerns. "Privacy Risks in Intelligent User Interfaces," from *IEEE Internet Computing*, reviews these concerns and ways to address them.

Applications developed with the popular C programming language can suffer buffer overflows. The authors of *IT Professional*'s "Defeating Buffer Overflow: A Trivial but Dangerous Bug" present some ways to detect and prevent this problem.

Cloud computing is increasingly being seen as a way to strengthen collaboration in manufacturing. However, security is a major concern with this approach. "Cloud Manufacturing: Security, Privacy, and Forensic Concerns," from *IEEE Cloud Computing*, looks into this issue.

The authors of *IEEE Security & Privacy*'s "The Outcomes of Cybersecurity Competitions and Implications for Underrepresented Populations" consider how these contests could increase student awareness of cybersecurity careers, focusing on gifted students and females, as well as low-income and high-risk groups.

ComputingEdge articles on topics other than cybersecurity include the following:

- The authors of *IEEE Software*'s "Creating the Virtual Universe" discuss a system they developed that includes an interface framework and 55 dedicated solvers for use with different kinds of physics problems.
- "Cloud Federation and the Evolution of Cloud Computing," from *Computer*, looks at the promise and challenge of interconnecting heterogeneous clouds to form federated systems that enable collective and collaborative cloud use.
- *IEEE Micro*'s "Congestion on the Last Mile" examines congestion that occurs when network capacity doesn't provide adequate service during heavy use.

2469-7087/17/$33.00 © 2017 IEEE

The Fog of War in Cyberspace

Alexander Kott, Ananthram Swami, and Bruce J. West,
US Army Research Laboratory

A "cyberfog" security approach that splits data into numerous fragments and continually disperses them across multiple end-user devices could provide greater attack resiliency but also presents formidable technical challenges.

The great Napoleonic Age warfare theorist, Carl von Clausewitz, wrote about the fog of war as the fundamental uncertainty of information in a complex and adversarial world. More recently, the term "fog computing" has emerged to refer to the extension of cloud computing to the network edge. We see connections between these seemingly disparate notions. For example, it might be possible to improve the security of our networks and data by maximizing the "fogginess" of information as it appears to a cyberadversary. Even if partly compromised, this information would remain opaque to the adversary, while still being useful to us.

One way to achieve such opaqueness is to split data into numerous fragments and continually disperse them across multiple end-user devices. Many modern commercial databases employ data splitting, or sharding, for both security and scalability, but typically not for end-user devices. However, given the growing interest in fog computing and fog networks,[1] and the maturing of edge-network distributed databases such as GaianDB[2] as well as cyber-physical networks, it's time to explore the use of data splitting at the edge.

While potentially offering numerous benefits such as greater attack resiliency, this "cyberfog" approach also presents formidable challenges with respect to data and network management complexity; bandwidth, storage, and battery-power demands; data-reassembly latency; and intermittent connectivity. In a recent meeting at the US Army Research Laboratory, government scientists discussed these challenges with colleagues from academia and industry.

DATA DISPERSION AND REASSEMBLY

The database security community has demonstrated, through both research prototypes and successful products, the feasibility and value of data dispersion and, to a lesser extent, frequent repositioning of data shards.[3]

2469-7087/17/$33.00 © 2017 IEEE

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

File confidentiality and integrity can be preserved, even when a cyberattack compromises a subset of the file servers.

Shamir's Secret Sharing scheme[4] can be seen as either a metaphor, or an actual component, of a cyberfog approach. Roughly, a Shamir-like data-dispersion scheme could enable information sharing in such a way that an adversary who succeeds in capturing a significant fraction of shards still won't be able to reconstruct any meaningful information from it. Such a scheme might help balance data-dispersion bandwidth requirements over time—for example, the bulk of data shards could be distributed during a lull in communications demands, whereas only the final and a few critical shards would be sent over the network during busy periods.

At the same time, there are significant obstacles to developing, validating, and implementing the complex mechanisms required to perform data dispersion. Increased diversification also creates new cyberattack surfaces and venues. In particular, a cyberfog approach could increase a network's vulnerability to availability attacks, even as it improves its resilience to confidentiality attacks. Consequently, the network might need to manage a complex tradeoff between availability and confidentiality in real time depending on users' tasks and circumstances. Achieving consistency would also be complicated.

Users eventually will request the dispersed data, which must be gathered and reassembled in a timely and efficient fashion. This could be helped by intelligent dispersion—putting data shards where they're more likely to be accessible when users are more likely to need them. While doing so, care must be taken not to introduce regularity into the dispersion scheme that would make it easier for adversaries to find that information. For example, CYRUS (Client-defined privacY-protected Reliable cloUd Service)[5] ensures user privacy and reliability by scattering files into smaller pieces across multiple clouds, so that no one cloud can read users' data.

To determine a user's data needs, there must be some means to automatically determine the relevance of information to the user. A cyberfog approach complicates this process: whereas in a conventional system two files in the same folder are likely relevant to the same issue, colocation of two data shards says nothing about their common relevance.

Timing issues in data dispersion and reassembly are also complex: the way a collection of information is dispersed—the data shards' size and distance from one another—depends on when and how rapidly the user will need these bundles of information, and the overhead for distributing and gathering each shard. The tradeoff between timeliness and security is dependent on the nature of the task: if maximum security need only be maintained for a short period of time, it might be acceptable that an adversary has a higher chance of obtaining the information after a given time interval. Researchers have explored placing data fragments and replicas so as to minimize latency in a dynamic disruption-tolerant network, taking into account users' social network structures.[6,7]

With a cyberfog approach, the network might need to manage a complex tradeoff between availability and confidentiality in real time depending on users' tasks and circumstances.

The network's topology, architecture, communication protocols, and other characteristics also influence the optimal means of data dispersion and reassembly. Fogging/defogging must take into account the size, density, complexity, and tempo of the network, the mobility and geographic proximity of users and nodes where data shards are stored, how soon sharded information will become stale, how soon stored information might be needed, and so on.

SITUATIONAL AWARENESS AND INFORMATION SEMANTICS

The ultimate goal of information accessibility is situational awareness (SA), and even timely and relevant information delivery doesn't guarantee high-quality SA. Not all data shards are equally valuable from the SA perspective: a given shard could be used to create multiple pictures or draw multiple conclusions, depending on how it's "glued" to other shards. SA thus presents a challenge with respect to discovering as well as gathering dispersed information.

A cyberfog approach will require novel methods of information fusion to achieve adequate SA, especially when data gathering is incomplete due to an adversary action or network failures. This entails knowledge of the semantic context of the information, which strongly influences how recipients understand it. Toward this end, semantic information theory[8] and perhaps sheaf theory[9] seem highly relevant to addressing cyberfog challenges.

Context is particularly important in protecting business tasks because an adversary might need very little information to disrupt a key element of a task. Consequently, the data-dispersion, data-gathering, and SA-formation processes must be designed and executed in such a way that information has high value for the users and low value for the adversary. This implies the need for a thorough model of the adversary's intent and prior knowledge.

RISK ASSESSMENT

Risk could serve as a comprehensive framework for characterizing cyberfog effectiveness. However, new risk models are needed to model poorly understood phenomena such as obfuscation that play an important role in this approach.

It's tempting to formulate the risk of failure in terms of data, such as the fraction of data captured by an adversary, but it should be analyzed in terms of the impact on a given task's objectives. This implies the need for an accurate model of the task, including its dependencies on network and computing assets—a highly complex modeling problem.

Other complexities arise in quantifying the impact of failure: the same failure can have very different consequences depending on its timing or how old the lost information is—the loss of dated information could be less important than that of recently obtained information. Additive properties of failures are important too—for example, knowing data item A and data item B might have high value, whereas knowing only one of the items would have zero value. The risk of a cyberfog approach also increases with the uncertainty of failure: if I know I lost data item A, I can do something about it; but if I'm uncertain, the approach's effectiveness is lessened.

Risk assessment in a cyberfog strategy would clearly benefit from a game-theoretic treatment. In this case, risk is highly dependent on the decisions and actions of the opponents, who are interdependent. This kind of game deviates strongly from the traditional zero-sum game because participants operate with partial information, bounded rationality, and so on. In fact, even the game's goals—the task's objectives—can be subject to change if some supporting assets fail or are captured by the adversary. Further, the game involves deception and obfuscation.

DECEPTION AND OBFUSCATION

Data dispersion presents adversaries with uncertainty as to where to find relevant information and how to reconstruct it from captured shards. A cyberfog approach also uses obfuscation and deception to increase uncertainty for the adversary. Obfuscation subjects information to multiple, equally possible interpretations, whereas deception aims to induce an incorrect interpretation that thwarts the adversary's goals. Obfuscation and deception can be achieved in many ways—for example, by providing a misleading view of the network's topology, traffic, and behavior.

Regardless of the means employed, effective obfuscation and deception can be difficult to implement. For example, creating believable fake business documents or network traffic is very challenging. The task is even harder if an adversary is able to observe network behavior and system use across both the physical and cyber dimensions.

Determining the fundamental limits of adversaries' ability to detect obfuscation and deception is also challenging. Because these are human fabrications, they're likely to be far less complex and rich in detail than real-world activities. As such, they might be vulnerable to sophisticated machine-learning techniques designed to detect anomalies. Thus, research is needed on ways to fool particular classifiers with particular inputs.[10] As AI systems become pervasive and increasingly sophisticated, understanding the difference between how machines and humans perceive obfuscation and deception will be critical to cyberfog success.

Given the extreme challenges and complexities inherent in a cyberfog environment, the use of formal methods could provide some assurance that the environment as well as the tools and activities we design for it exhibit certain properties. Unfortunately, formal methods are expensive to implement and can't yet eliminate the need for conventional testing. Nor are formal methods suitable for novel types of cyberattacks that contravene current models' assumptions.[11] Furthermore, it's unknown how well, if at all, formal methods apply to human factors such as the role of cognition in deception. Perhaps some of these difficulties could be mitigated by purposefully designing a cyberfog strategy that's more amenable to formal methods. C

REFERENCES

1. M. Chiang, "Fog Networking: An Overview on Research Opportunities," Dec. 2015; arxiv.org/pdf/1601.00835.pdf.
2. G. Bent et al., *Network and Information Sciences International Technology Alliance*, US Army Research Lab/UK Ministry of Defence, 2016; nis-ita.org/Legacy/files/book/ITA%20eBook%20PDF.pdf.
3. A. Mei, L.V. Mancini, and S. Jajodia, "Secure Dynamic Fragment and Replica Allocation in Large-Scale Distributed File Systems," *IEEE Trans. Parallel and Distributed Systems*, vol. 14, no. 9, 2003, pp. 885–896.
4. A. Shamir, "How to Share a Secret," *Comm. ACM*, vol. 22, no. 11, 1979, pp. 612–613.
5. J.Y. Chung et al., "CYRUS: Towards Client-Defined Cloud Storage," *Proc. 10th European Conf. Computer Systems* (EuroSys 15), 2015; www.princeton.edu/~cjoe/CYRUS_EuroSys.pdf.

This article originally appeared in Computer, *vol. 49, no. 11, 2016.*

6. X. Zhuo et al., "Social-Based Cooperative Caching in DTNs: A Contact Duration Aware Approach," *Proc. IEEE 8th Int'l Conf. Mobile Ad-Hoc and Sensor Systems* (MASS 11), 2011, pp. 92–101.
7. W. Gao et al., "Cooperative Caching for Efficient Data Access in Disruption Tolerant Networks," *IEEE Trans. Mobile Computing*, vol. 13, no. 3, 2014, pp. 611–625.
8. P. Basu et al., "Preserving Quality of Information by Using Semantic Relationships," *Pervasive and Mobile Computing*, vol. 11, 2014, pp. 188–202.
9. G.E. Bredon, *Sheaf Theory*, 2nd ed., Springer, 1997.
10. P. McDaniel, N. Papernot, and Z.B. Celik, "Machine Learning in Adversarial Settings," *IEEE Security & Privacy*, vol. 14, no. 3, 2016, pp. 68–72.
11. K. Schaffer and J. Voas, "What Happened to Formal Methods for Security?," *Computer*, vol. 49, no. 8, 2016, pp. 70–79.

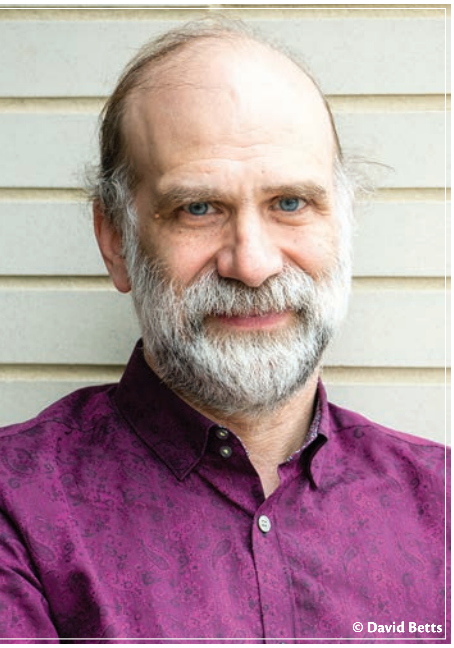
DISCLAIMER

This article doesn't reflect the positions or views of the authors' employers.

ALEXANDER KOTT is chief of the Network Science Division, US Army Research Laboratory. Contact him at alexander.kott1.civ@mail.mil.

ANANTHRAM SWAMI is senior research scientist for network science at the US Army Research Laboratory. Contact him at ananthram.swami.civ@mail.mil.

BRUCE J. WEST is senior scientist in mathematics at the Information Sciences Directorate, Army Research Office, US Army Research Laboratory. Contact him at bruce.j.west.civ@mail.mil.


© David Betts

Bruce Schneier
Harvard University

This article originally appeared in IEEE Security & Privacy, *vol. 14, no. 5, 2016.*

Stop Trying to Fix the User

Every few years, a researcher replicates a security study by littering USB sticks around an organization's grounds and waiting to see how many people pick them up and plug them in, causing the autorun function to install innocuous malware on their computers. These studies are great for making security professionals feel superior. The researchers get to demonstrate their security expertise and use the results as "teachable moments" for others. "If only everyone was more security aware and had more security training," they say, "the Internet would be a much safer place."

Enough of that. The problem isn't the users: it's that we've designed our computer systems' security so badly that we demand the user do all of these counterintuitive things. Why can't users choose easy-to-remember passwords? Why can't they click on links in emails with wild abandon? Why can't they plug a USB stick into a computer without facing a myriad of viruses? Why are we trying to fix the user instead of solving the underlying security problem?

Traditionally, we've thought about security and usability as a tradeoff: a more secure system is less functional and more annoying, and a more capable, flexible, and powerful system is less secure. This "either/or" thinking results in systems that are neither usable nor secure.

Our industry is littered with examples. First: security warnings. Despite researchers' good intentions, these warnings just inure people to them. I've read dozens of studies about how to get people to pay attention to security warnings. We can tweak their wording, highlight them in red, and jiggle them on the screen, but nothing works because users know the warnings are invariably meaningless. They don't see "the certificate has expired; are you sure you want to go to this webpage?" They see "I'm an annoying message preventing you from reading a webpage. Click here to get rid of me."

Next: passwords. It makes no sense to force users to generate passwords for websites they only log in to once or twice a year. Users realize this: they store those passwords in their browsers, or they never even bother trying to remember them, using the "I forgot my password" link as a way to bypass the system completely—effectively falling back on the security of their email account.

And finally: phishing links. Users are free to click around the Web until they encounter a link to a phishing website. Then everyone wants to know how to train the user not to click on suspicious links. But you can't train users not to click on links when you've spent the past two decades teaching them that links are there to be clicked.

We must stop trying to fix the user to achieve security. We'll never get there, and research toward those goals just obscures the real problems. Usable security doesn't mean "getting people to do what we want." It means creating security that works, given (or despite) what people do. It means security solutions that deliver on users' security goals without—as the 19th-century Dutch cryptographer Auguste Kerckhoffs aptly put it—"stress of mind, or knowledge of a long series of rules."

I've been saying this for years. Security usablity guru (and one of this issue's guest editors) M. Angela Sasse has been saying it even longer. People—and developers—are finally starting to listen. Many security updates happen automatically so users don't have to remember to manually update their systems. Opening a Word or Excel document inside Google Docs isolates it from the user's system so there's little risk of embedded malware. And programs can run in sandboxes that don't compromise the entire computer. We've come a long way, but we have a lot further to go.

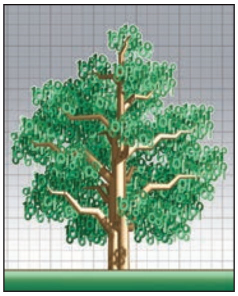
Blame-the-victim thinking is older than the Internet, of course. But that doesn't make it right. We owe it to our users to make the Information Age a safe place for everyone—not just those with "security awareness." ■

Bruce Schneier is a security technologist at the Berkman Klein Center for Internet and Society at Harvard University. He's also the CTO of Resilient and Special Advisor to IBM Security. Contact him via www.schneier.com.

2469-7087/17/$33.00 © 2017 IEEE

Editor: Munindar P. Singh • m.singh@ieee.org

Privacy Risks in Intelligent User Interfaces



Christopher J. Hazard • *Hazardous Software*

Munindar P. Singh • *North Carolina State University*

Intelligent user interfaces (in games, for example) provide opportunities for producing a high-quality, contextually relevant user experience. However, they also raise the specter of privacy violations. The authors review some of the ways in which user interfaces could glean a user's private information; then the authors highlight the risks therein, and discuss ways of mitigating those risks.

We define an *intelligent user interface* or IUI as (part of) an app that interacts with a user in a way that's responsive to the user's changing needs at the time of interaction. That is, an IUI provides functionality in a way that is adaptive to specific users and to their specific contexts of usage, as those contexts arise and change in the field. Typically, an IUI would construct and maintain a model of the user and the user's context. As part of doing so, an IUI would capture relevant aspects of the user's profile (possibly including demographic information), interaction and communication history, goals, preferences, social relationships, traits such as personality, and physiological and psychological states. Not every IUI needs all these aspects, but depending upon the underlying purpose of the app and how ambitious its designers are, an IUI might capture more or fewer of them.

Examples of IUIs include tools that support calendars and navigation (such as Google Now); dialogue apps (such as Apple's Siri); and games — both those on fixed devices and those that are inherently mobile (such as Pokémon Go).

IUIs can function effectively only because of the information they collect or access about each user. Some information might be provided by any of the following:

- directly from the user (such as your age and sex);
- user-allowed access to other services (such as your email content, friend lists, and such);
- explicit interactions with the user (such as through your prior queries and their results);
- data implicitly gathered about the user (such as from the locations you visited or the locations where you played a particular game);
- explicit requests from the IUI (for example, if it asks whether you would you like to receive this call); and
- inferred user interactions (such as your preference for less-interactive content during the morning and late at night).

Armed with this information, an IUI seeks to offer an enhanced user experience by figuring out the user's goals and preferences and acting accordingly.

Under weak assumptions of how users behave or by learning such patterns across the entire body of users, an IUI can figure out additional details about a user that the user might never have realized were being revealed. For example, it isn't farfetched to guess that a user's home or work is one of the locations at which a user is most frequently present or one of the origin locations from where the user most often searches for routes to other locations.[1] In addition, users (even those who work and live in the same locations) would have mutually distinct trajectories on a day-to-day basis — thus, users' trajectories can serve as pseudonyms for them.

Increasingly, privacy is recognized as a major concern. The privacy risks of games have received public and congressional attention (see www.franken.senate.gov/files/letter/160712_PokemonGO.

2469-7087/17/$33.00 © 2017 IEEE Published by the IEEE Computer Society

pdf). As IUIs collect more types and amounts of data on users, the associated risk of disclosure increases. Moreover, privacy is more than a concern about access to information; it includes considerations such as infringement on a person's autonomy, intrusion into private space, and loss of dignity.[2] A proper understanding of privacy not only can help us reduce avoidable risks, but by doing so, also reduce the so-called "chilling effect" of government or corporate surveillance on people's behaviors, and thereby enhance the potential individual and societal value of modern intelligent apps.

Why IUIs Are on the Rise: Potential Benefits

IUIs are expanding because they're valuable. As the available information and decisions grow, there's an increasing need to select appropriately among them. In addition, user time, attention, and effort are increasingly at a premium as information technology is deployed in more and more natural settings, not merely in your office. As a result, users do need greater support in their decision making, and such support must accommodate the user's needs by taking into account a rich model of the user.

In simple terms, what IUIs offer is intelligent discrimination between numerous raw possibilities to select actions that best capture a user's goals. For example, if the authors (based in Raleigh, North Carolina) are looking for an address in Durham, they more likely mean Durham, North Carolina and not Durham, United Kingdom. A navigation app that automatically chooses the nearby Durham can do so only if it knows where the requestor is based. We wonder if an IUI would have helped avoid the error that led a Belgian woman on a 3,000 km off-course drive.[3]

The problem requires greater intelligence than fixed rules, however. For example, if an email indicates an airline ticket booked to Tees Valley Airport, then maybe it's the UK's Durham that's salient, though with the origin set to Tees Valley Airport.

Likewise, a game or an educational app might choose between challenges to present to a user based on how tired or competent the user is — better players or students get harder challenges so they won't be bored and others get simpler challenges so they won't be frustrated. This is nothing but an application of Mihaly Csikszentmihalyi's[4] idea of the flow channel, and is a commonplace tenet in game design.[5] Of course, to support such functionality presupposes determining how competent, tired, or anxious the user is.

Privacy Risks

In a nutshell, IUIs bring forth the following tension: To operate effectively, they need to acquire or construct rich information about the user. The most valuable of such information is potentially sensitive and revealing; it can pose a threat to the user's safety, finances, or dignity (just imagine if it becomes known that you're the slowest student in your class).

Privacy risks arise in a variety of settings. For example, if you stored a "home" location on your navigation app on your phone, a criminal who steals your phone can then navigate to your home as well to rob or attack you. We don't emphasize such risks in this article, because they rely upon an external attack on an IUI or a device. Instead, we primarily consider risks where the attack is through the app itself. An example of such an attack would be where your navigation app routes you by an ice cream shop or a pub on your way home, based on the assumption that a subtle suggestion (when you're tired at the end of a long day of walking or driving) might cause you to visit such an establishment.

Extracting and Disseminating Information

Information can be extracted from machine learning models that have been trained, even if the original data isn't accessible.[6] Such models function as a form of data compression of a subset of the user's private data, capturing the nuggets of information that are potentially most sensitive for the user. In many cases, the user might not have known that sensitive data was being collected, because it's hidden within routine data, but machine learning brings it forward. For example, consider an IUI that learns a user's preferences over time for the purpose of improving user productivity. In this example, the IUI might learn artifacts about the user that aren't explicitly related to the task, such as the time that the user wakes up in the morning or what times of day the user isn't productive. Neither the user nor the app developer might have realized that this information was contained within the learned data.

If the IUI were to disclose such sensitive information to others, that would be a privacy risk. For example, if your calendar informed your boss that you began work not at 8:00 a.m. but at 10:00 a.m., that might be significant. The outcome might be just as harmful if the calendar informed your clients that you were available at 8:00 a.m. but not ready to talk to them, simply because you were reserving the time for "more important" tasks.

Probing Users

An IUI doesn't merely have to passively observe a user; it can actively probe a user by presenting carefully chosen alternatives to a user as a way to learn about the user's physiological or psychological state. From the choices the user makes, an IUI can potentially infer information about whether the user is depressed[7] or dieting,[8] and can estimate other psychographic measures related to decision fatigue. Recent work has suggested that decision fatigue and ego depletion may be at least somewhat specious[9] (or at least not reliably reproducible), calling some question on the validity of some previous studies. However, a widely deployed app that performs empirical analysis doesn't have to work in general, only in its particular setting. Such an app can quickly gather actionable empirical results far larger than academic studies, possibly incentivizing the developer to keep the data proprietary for commercial gain. If an IUI can, in some way, utilize some aspect of decision fatigue, the user can be controlled in unusual ways.

Compromising Security and Identity

Many authentication protocols rely upon bringing out shared secrets. For example, credit card transactions often require stating the customer's home address to corroborate that the customer is legitimate. And, when a situation raises some red flags, credit card companies ask users to verify which transactions they carried out at which sites — presuming that only the genuine user would know of them. But a location-based app might be able to guess your home address as well as brick-and-mortar establishments you've visited where you might have made purchases. So a rogue IUI can easily help compromise your security and identity.

Directly Manipulating Autonomy

We define *direct manipulation of autonomy* as partial or total control over a user's actions characterized by a moderate to high probability of success for any given interaction. In other words, it's likely that a user experiencing this form of manipulation will have a high likelihood of being coerced into doing something they otherwise wouldn't have done. These types of manipulations might or might not require private information to work, but they might be enhanced by private data or personally identifiable information (better known as PII) and they could yield private data or PII.

Dark patterns, wherein a user interface is crafted to trick users into performing a particular task, are instances of attacks that directly manipulate autonomy (see http://darkpatterns.org). An example of a dark pattern is a navigation app that repeatedly asks, until you agree, if you would like to permanently allow the service provider to collect detailed data from your phone to improve your results. By frustrating the user enough, such an app in effect coerces the user to agree after a few episodes: subsequently, the user might forget having granted this permission or be unable to find a way to rescind the permission.

Indirectly Manipulating Autonomy

We define *indirect manipulation of autonomy* as partial control over a user's actions, characterized by an extremely low probability of success in manipulation at any given interaction. In other words, a successful manipulation either requires exposure to a large audience, numerous exposures to the same user, or both.

Examples of indirect manipulation of autonomy are advertisements, layouts of interfaces, hardware, or other interactions that yield slight differences in behavior in aggregation. Changes in interfaces, for example, relate to what's called *choice architecture*,[10] where the choices being encouraged are given prominence or made easier. For example, many casual games have in-game purchases that allow the player to advance more quickly through difficult or frustrating parts of the game. The game developer can present the player the option to purchase an item that will increase the chances of speeding through the difficult section at the most opportune times. By gathering data en masse about players, various analytical techniques can indicate when players are most likely to make a purchase and how to improve retention when players are about to stop playing the game, enabling developers to capitalize on these tendencies.

Aggregate data about individuals can drive indirect manipulation of autonomy by giving those who employ such information means to measure, classify, and segment their target audiences while empirically testing the results of their indirect manipulations.

Casual mobile games exemplify an IUI that could deplete self-control, increase cognitive load, and present the user with the option to make decisions against their better judgment. Popular games — such as Clash of Clans, Candy Crush Saga, and Pokémon GO — feature numerous decisions that each seem vitally important yet don't generally alter the long-term course of a player's experience. Although game developers generally seek to increase revenue by improving the user's experience,[11] a deceitful actor could apply such techniques to exploit a user by presenting decisions precisely at times when the user is at a disadvantage.

Prospects for Mitigation

How can we mitigate the foregoing risks without losing the benefits of IUIs?

Ethical IUI Design

A straightforward approach is to push for stronger standards for ethics among content and service providers who create or utilize IUIs. A combination of industry standards, social norms, legislation, Institutional Review Board (IRB) practices, and certifications could mitigate some privacy concerns when deploying commercial services. Although some developers of IUIs consider complex ethical matters,[12] privacy doesn't have ubiquitous support due to numerous cultural factors that can make privacy appear to be a minor concern.[13]

Architectural Solutions and Open Standards

Sound architecture and algorithms can enhance privacy while allowing providers access to the data and analytics needed in IUIs. Differential privacy guarantees protection in some situations by adding noise or resampling data.[14] Contextual middleware[15] provides a high-level API to IUI apps that hides user-specific sensor data and reveals only the user's readiness for an intelligent action by the app. These two approaches could be adapted for IUIs by weakening the connection between the decisions needed in a game and the user's state.

User Agents

User agents — originating in a trusted operating system or device, and which reflect the user's interests — can help a user cope with privacy threats from IUIs. Similar techniques have proved valuable for low-level aspects, such as browser fingerprinting (for example, see Secret Agent; www.dephormation.org.uk/?page=81). Here we have in mind agents that accommodate richer models of threats to users than mere traceability of actions.

Agents could filter input data on the front end or notify a user when there's an increased risk of compromising sensitive information. For example, an agent could determine which data fields are necessary for a service and which are risky given the user's interests. An agent could provide correct data for legitimate purposes (the address needed for shipping) and fill in randomized data to enhance privacy in other cases (randomizing birthdates, for example, without affecting determination of adulthood).

Agents could filter on the back end, by monitoring content transmitted and API calls, such as Android and iOS support app permissions. Or the agent could act as a content-aware firewall and analyze and filter data before it's sent to the service provider. If a game is sending a user's contact list to a third party, such an agent could block the content from being sent.

Agents presuppose an open architecture. Given technological and legal ways — "walled gardens" — by which platform and content vendors restrict users' ability to automatically interact with software,[16] such agents might not be viable. This situation only highlights the need for openness, possibly through government regulation.

Economic Models

Defending yourself in an environment that includes hostile agents or contentious resources often requires nontrivial resource expenditure, or at least signaling a commitment to expend nontrivial resources, regardless of the domain. A person's private information and identity are valuable in many contexts, and IUIs are a key component in the arms race between privacy and exploitation, and between different vested interests, such as service providers and ad blockers.[17]

Game-theoretic approaches, which concern strategies of competing players (here, IUI providers and users), can help develop mechanisms that optimize some objective. We conjecture that techniques developed to protect physical infrastructure[18] can be enhanced for IUIs.

Provenance and Auditability

If we can store how analyses and actions are derived from some data, we can verify whether the data were used in a way unintended by the user. Blockchain technologies provide a way to store data (typically publicly) such that only holders of a cryptographic key can compute on and validate the content. Potentially, privacy-preserving blockchain contracts[19] might be extended to support a provenance mechanism, such that any transaction or analysis that depends on any other data could indicate which data it depends on without giving away the content.

As illustration, suppose an IUI provider is contractually bound to explain its decisions. That is, it might use personal data about users, but must store all analyses and decisions in a blockchain with references to specific data from which it was derived. A user could audit the blockchain to verify if any of his or her data was used for purposes outside the contract's scope. Tools would help perform the audit. This approach, however, is far from perfect. The relationships between the private data stored in the blockchain could reveal sensitive information about the user or trade secrets of the IUI provider.

People often find manipulation to be one of the most egregious personal violations — witness the controversy over Facebook's newsfeed manipulation.[20] Although manipulation might not involve information disclosure, it violates privacy by attacking a person's dignity. Because identity and integrity of autonomy are key to a person's sense of self, IUIs not only reveal a large attack surface but also expose particularly insidious risks. Understanding and addressing such risks is crucial for the future advancement of IUIs.

Improved methods are needed to help mitigate privacy risks, to balance privacy and utility. Methods involving architectures and agents are closest to practice; ideas from auditability and economics show promise as well.

Acknowledgment

Munindar Singh thanks the US Department of Defense for support through the Science of Security Lablet.

References

1. R. Liu et al., "An Unsupervised Collaborative Approach to Identifying Home and Work Locations," *Proc. 17th IEEE Int'l Conf. Mobile Data Management*, 2016; doi:10.1109/MDM.2016.53.
2. W.L. Prosser, "Privacy," *California Law Rev.*, vol. 48, no. 3, 1960, pp. 383–423.
3. Yahoo, "Belgian Woman Drives 3000km across Europe by Mistake," *Yahoo.com*, 16 Jan. 2013; https://nz.lifestyle.yahoo.com/travel/a/15850672/belgian-woman-drives-3000km-across-europe-by-mistake.
4. J. Nakamura and M. Csikszentmihalyi, "The Concept of Flow," *Handbook of Positive Psychology*, C.R. Snyder and S.J. Lopez, eds., Oxford Univ. Press, 2002, pp. 89–105.
5. T. Sala, "Game Design Theory Applied: The Flow Channel," *Gamasutra*, 8 Dec. 2013; www.gamasutra.com/blogs/ToniSala/20131208/206535/Game_Design_Theory_Applied_The_Flow_Channel.php.

6. P. Cortez and M.J. Embrechts, “Using Sensitivity Analysis and Visualization Techniques to Open Black Box Data Mining Models,” *Information Sciences*, vol. 225, 2013, pp. 1–17.
7. C.W. Korn et al., “Depression Is Related to an Absence of Optimistically Biased Belief Updating about Future Life Events,” *Psychological Medicine*, vol. 44, no. 03, 2014, pp. 579–592.
8. K.E. D’Anci, “Reduced-Calorie Diets and Mental Performance in Adults,” *Nutrition and Mental Performance*, ch. 10, Macmillan, 2012, pp. 179–193.
9. M.S. Hagger and N.L.D. Chatzisarantis, “A Multi-Lab Pre-Registered Replication of the Ego-Depletion Effect,” *Perspectives on Psychological Science*, vol. 11, no. 4, 2016, pp. 546–573.
10. C.R. Sunstein, *The Ethics of Nudging*, tech. report 2526341, Social Science Research Network, 2014; http://dx.doi.org/10.2139/ssrn.2526341.
11. J. Newman, J. Jerome, and C.J. Hazard, “Press Start to Track Privacy and the New Questions Posed by Modern Video Game Technology,” *Am. Intellectual Property Law Association (AIPLA) Quarterly J.* vol. 42, no. 4, 2014, p. 527; www.aipla.org/learningcenter/library/books/qj/Pages/Quarterly-Journal-42-4.aspx.
12. V. Koenig, F. Boehm, and R. McCall, “Pervasive Gaming as a Potential Solution to Traffic Congestion: New Challenges Regarding Ethics, Privacy, and Trust,” *Proc. Int’l Conf. Entertainment Computing*, LNCS 7522, Springer, 2012, pp. 586–593.
13. S. Cockcroft and S. Rekker, “The Relationship between Culture and Information Privacy Policy,” *Electronic Markets*, vol. 26, no. 1, 2016, pp. 55–72.
14. C. Dwork and A. Roth, “The Algorithmic Foundations of Differential Privacy,” *Foundations and Trends in Theoretical Computer Science*, vol. 9, nos. 3–4, 2014, pp. 211–407.
15. P.K. Murukannaiah and M.P. Singh, “Platys: An Active Learning Framework for Place-Aware Application Development and Its Evaluation,” *ACM Trans. Software Engineering and Methodology*, vol. 24, no. 3, 2015, pp. 19:1–19:32.
16. T. Sweeney, “Microsoft Wants to Monopolise Games Development on PC. We Must Fight It,” *The Guardian*, 4 Mar. 2016; www.theguardian.com/technology/2016/mar/04/microsoft-monopolise-pc-games-development-epic-games-gears-of-war.
17. J. Constine, “Facebook Rolls out Code to Nullify Adblock Plus’ Workaround Again,” *Tech Crunch*, 11 Aug. 2016; https://techcrunch.com/2016/08/11/friendblock/.
18. M. Tambe, *Security and Game Theory*, Cambridge Univ. Press, 2011.
19. A. Kosba et al., “Hawk: The Blockchain Model of Cryptography and Privacy-Preserving Smart Contracts,” *Proc. IEEE Symp. Security and Privacy*, 2016, pp. 839–858.
20. G.S. McNeal, “Controversy Over Facebook Emotional Manipulation Study Grows as Timeline Becomes More Clear,” *Forbes*, 30 June 2014; www.forbes.com/sites/gregorymcneal/2014/06/30/controversy-over-facebook-emotional-manipulation-study-grows-as-timeline-becomes-more-clear/#5de2f38a4e44.

Christopher J. Hazard is the founder of Hazardous Software (a game company known for the award-winning 2011 strategy game Achron). His work spans a variety of fields, including AI, trust and reputation, networks, cybersecurity, robotics, psychology, privacy, economics, and logistics. Hazard has a PhD in computer science from North Carolina State University. Contact him at cjhazard@hazardoussoftware.com.

Munindar P. Singh is a computer science professor at North Carolina State University. His research interests include the conception, engineering, and governance of sociotechnical systems as a way to tackle concerns such as security and privacy. Singh is an IEEE Fellow, a former Editor in Chief of *IEEE Internet Computing*, and the current Editor in Chief of *ACM Transactions on Internet Technology*. Contact him at singh@ncsu.edu.

This article originally appeared in IEEE Internet Computing, *vol. 20, no. 6, 2016.*

EDITORS: Rick Kuhn, US National Institute of Standards and Technology, kuhn@nist.gov
Tim Weil, Scram Systems, tweil.ieee@gmail.com

Defeating Buffer Overflow

A Trivial but Dangerous Bug

Paul E. Black and Irena Bojanova, *US National Institute of Standards and Technology*

The C programming language was invented more than 40 years ago. It is infamous for buffer overflows. We have learned a lot about computer science, language design, and software engineering since then. Because it is unlikely that we will stop using C any time soon, we present some ways to deal with buffer overflow. Many of these techniques are also useful for other programing languages and other classes of vulnerabilities.

Definition and Description

The term "buffer" comes from decades ago when I/O operations were slow. Memory was set aside to hold a chunk of output data going to a device—such as a printer or a 1,200 bit/s modem—or input data being received from a keyboard or a punch card reader. When the buffer access was finished, the computer was interrupted to set up another I/O operation. The term has come to mean a chunk of contiguous memory whose values constitute a larger whole. For instance, a string is often stored as characters kept in a contiguous set of memory locations. We use the C language standard term "array," but retain the common, although less precise term "buffer overflow."

An array is a semistructured group of elements of the same type. The elements are accessed by integer indexes. In C, arrays are zero-based—that is, the first element has index 0. Other languages are one-based or allow the user to define the first index. In C, valid indexes range from zero to the total number of elements, minus one. Because C allows a reference (pointer) into an array, an indexed access with a negative index might be valid, too.

The Bugs Framework (BF) defines the *buffer overflow* (BOF) class as follows: "The software accesses through an array a memory location that is outside the boundaries of that array."[1] In other words, the program uses an array reference to read from or write to a memory location that is before the beginning or after the end of the array. The BF provides information on the causes, attributes, and consequences of other bug classes, such as *injection* (INJ), *information exposure* (IEX), and *control of interaction frequency* (CIF).

Figure 1 shows that there are only two proximate causes of BOF: *data exceeds array* (that is, the amount of data exceeds the size of the array), or there is a *wrong index* or *pointer out of range*. These might be a result of other causes, too. *Data exceeds array* has two specific cases. In the first case, the programmer allocated the *array too small*, as in CVE-2015-0235–Ghost (https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-0235). The code computes the size of the needed array but leaves out one factor, which makes the array four bytes short. In the second case, *too much data* was accessed, as in CVE-2014-0160–Heartbleed (https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-0160). Instead of finding the length of the reply string that is already stored in an array, the code uses a number from an input. So, bad input can cause the code to read far too much data. The chain of causes for Heartbleed is *input not checked* *properly*, which leads to *too much data* read—specifically, a huge number of bytes are read from the heap.

Buffer overflow causes failures because data is read or written in ways that are entirely foreign to what the programmer plans. Memory contains information, such as the address of the next instruction to execute after returning from a function, calling parameters, variables used in the function, data structures, and permission flags set by the operating system. Writing outside an array could change any of these. In the worst-case scenario, adversaries could cause the program to gain extra permission or make the program execute arbitrary code. Reading beyond array boundaries could retrieve sensitive data, such as old passwords, that are left in memory after they are processed.

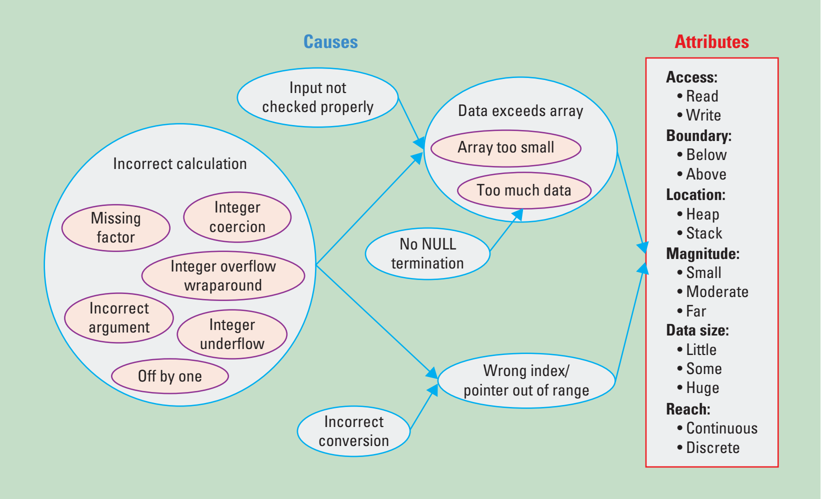


Figure 1. Buffer overflow (BOF) causes and attributes.[1]

Detecting Buffer Overflows

Buffer overflow can be detected through two general approaches: internal and external. Internal mechanisms are those that are built into a program and operate during execution. External or static mechanisms do not access the state of executing programs. The first external mechanism is observing a program's behavior.

Almost any failure could be the result of many kinds of bugs. However, some failures have characteristics strongly suggesting BOF as the software weakness:

- Is far more information produced than expected? This suggests a read BOF. Heartbleed might have been discovered earlier if we had verified that responses to heartbeat packets were only a few dozen bytes.
- Is different data corrupted in unusual ways in response to specific input? For instance, does a longer input cause a different failure than a shorter input? This suggests a write BOF.
- Does the program crash, and a dump or debugger give nonsensical stack traces? This suggests a write BOF of stack locations that corrupts the call/return stack.

Static analyzers check programs for possible BOF and other issues. Sound static analyzers are potentially always correct. In contrast, heuristic analyzers generally run faster, handle more languages, and cover more classes of vulnerabilities. Today, most static analyzers have lower false-positive rates and simultaneously lower false-negative rates than they had five years ago. Some static analyzers have been augmented with execution monitoring to yield hybrid (static and dynamic) analyzers.

Good general testing techniques complement static analysis. Testing relies on fewer assumptions and checks properties that are difficult to specify. We mention a few points particularly important to testing for BOF:

- Try to exceed limits, check routines that allocate more memory, and challenge the limits of hard-coded arrays.
- Try very unusual inputs, such as negative numbers, empty fields, and letters or special symbols where numbers are expected.

In contrast to the aforementioned external methods, internal detection mechanisms have access to the program's state and control flow. Many of them not only detect BOF but also help prevent failures or lessen their impact. Therefore, in the next section, we include internal ways to mitigate or preclude BOF with the discussion of ways to internally detect them.

Internal Detection and Prevention

The best way to prevent BOF is to reduce the use of C. Optimizing compilers and multicore processors remove most concerns about slower execution, allowing programmers to work on algorithmic improvements instead of checking every array access for a possible BOF. If you must write in C, use more structured stores, such as associative memory, prop lists, graphs, queues, sets, stacks, or trees. These abstract data structures bundle accessing operations that allow access only to valid elements. Arrays have minimal structure: just the index order.

There are many internal techniques for detecting, mitigating, or precluding BOF faults. They are either passive (detect that BOF has occurred) or active (prevent BOF). Also, they either require a programmer's action in order to be inserted or are inserted automatically by the compiler or the OS. One technique is to add checks to verify that every access is within bounds. Research shows that many bounds-checking tools or libraries have little impact on speed.[2] Chips with multiple, deeply pipelined cores can check bounds while the array is being accessed. Checking can also be done by the hardware. For instance, arrays might have read-only or unallocated blocks of memory on both ends. Small invalid array accesses result in memory violation interrupts. If performance still suffers, such checking could be enabled during development and testing, then disabled for production.

Some of these techniques might not be applicable—for example, if the size of the buffer is not available to check. In such cases, more sophisticated techniques attempt to foil adversaries.

Shadowing and fat pointers keep additional information about memory use and allocation in other parts of memory to enable access and taint checks.[3] Address space layout randomization (ASLR) distributes arrays unsystematically in memory. With ASLR, a BOF is unlikely to access the same unassociated object in different executions without a lot of work. Information that is connected, such as in the stack or in the same structure, is harder to rearrange. Padding allocates extra space for every array, so small magnitude BOF events might not cause problems. "Canaries" are special values, such as 0xDEADBEEF, added before and after arrays. If these values change, it is likely that a write BOF occurred.

Testing for Buffer Overflows

Testing for BOF is still crucial even when programs use good techniques. Test cases specifically targeted to exposing BOF can be generated through fuzzing, memory checking, and negative testing.

Fuzzing is a class of techniques in which random or structured random input is presented to a program with only limited checking of the outcome. Often, the only checking is that the program did not crash or hang. Because fuzzing automates input generation and output checking, huge numbers of tests can be run at little cost other than a few hours of computer time. Fuzz testing is powerful because random inputs expose the limits of programmers' analyses, or they violate assumptions about inputs that can never occur.

Structured random inputs are more powerful than purely random inputs, given that the latter primarily exercise the input checking routines. For instance, if a particular input is a date, it is useful to run only a moderate number of purely random tests. Any additional random tests are almost always handled by the code that tests whether the date is invalid. After a moderate number of tests, structured random dates can be generated with random months from 1 to 12, random days from 1 to 31, and a wide range of years. Another approach to structured random inputs is to capture known input and randomly mutate it. For instance, image display programs can be fed actual images with random changes.

Fuzzing with memory checking can be very effective. For instance, *american fuzzy lop* (afl) "tracks the branches that are taken and how often, then prefers using tests that cover the program differently when it evolves new tests" (http://lcamtuf.coredump.cx/afl).

Exact memory checkers, such as *Address Sanitizer* (ASan) or *Purify*, check memory allocation and layout. The overhead can be significant, up to twice the execution time and memory use, but this may be cheap insurance against vulnerabilities.

Negative testing examines how the program behaves when inputs are not as expected. The vast majority of testing is designed to gain confidence that the program produces expected outputs for typical inputs. As Wheeler says,

> *Thorough negative testing … creates a set of tests that cover every type of input that should fail. … This would have immediately found Heartbleed, since Heartbleed involved a data length value that was not correct according to the specification. It would also find other problems like CVE-2014-1266, the goto fail error in the Apple iOS implementation of SSL/TLS.*[4]

You do not have to suffer from BOF. Buffer overflows can cause serious problems, especially when we acknowledge the possibility of adversaries who try to exploit vulnerabilities in your programs. The best approach to ensuring that your software does not have buffer overflows is to use a programming language in which such bugs are impossible (memory access is always handled reliably) or, at least, can surely be detected by tools during production. There are many techniques that detect the vast majority of buffer overflows. There is no reason for your development process to be interrupted scrambling to patch them. IT

References

1. I. Bojanova et al., "The Bugs Framework (BF): A Structured Approach to Express Bugs," *Proc. 2016 IEEE*

Int'l Conf. Software Quality, Reliability and Security (QRS), Aug. 2016.

2. D. Flater, "Defensive Code's Impact on Software Performance," tech. note 1860, US Nat'l Inst. Standards and Technology, Jan. 2015; https://doi.org/10.6028/NIST.TN.1860.
3. E.D. Berger and B.G. Zorn, "DieHard: Probabilistic Memory Safety for Unsafe Languages," *Proc. 27th ACM SIGPLAN Conf. Programming Language Design and Implementation*, 2006, pp. 158–168; https://people.cs.umass.edu/~emery/pubs/fp014-berger.pdf.
4. D.A. Wheeler, "How to Prevent the Next Heartbleed," blog, 29 Apr. 2014; www.dwheeler.com/essays/heartbleed.html.

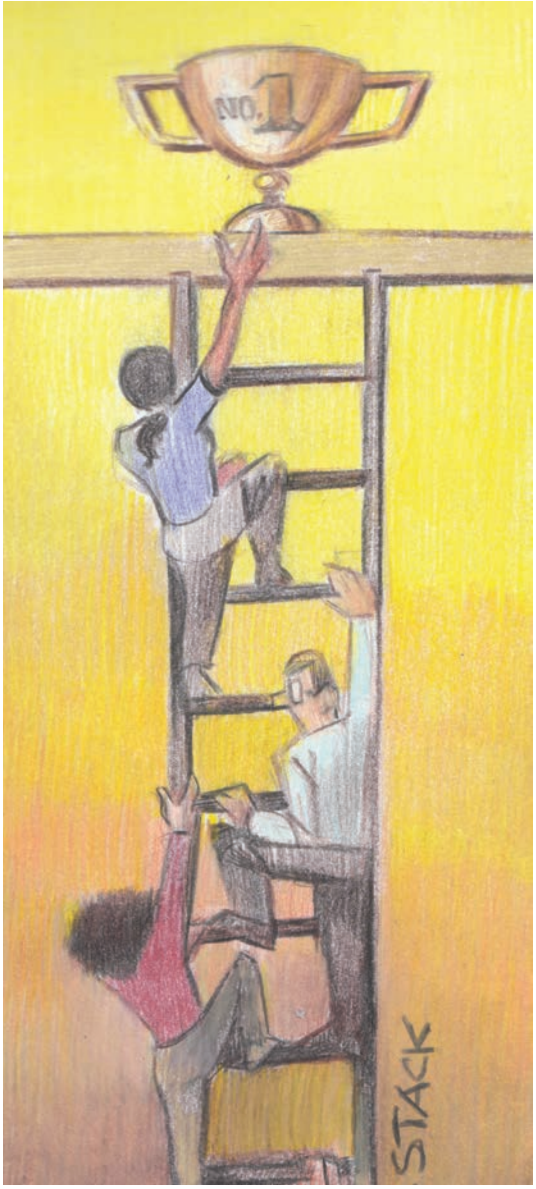
***Paul E. Black** is a computer scientist at the US National Institute of Standards and Technology. His research interests include static analysis, software testing, networks and queuing analysis, formal methods, software verification, quantum computing, and computer forensics. Black has nearly 20 years of industrial experience developing software for integrated circuit design and verification, assuring software quality, and managing business data processing. He is the editor of the* Dictionary of Algorithms and Data Structures *(www.nist.gov/dads/), and is a member of ACM and a senior member of IEEE. Contact him at paul.black@nist.gov.*

***Irena Bojanova** is a computer scientist at the US National Institute of Standards and Technology. She serves as the Committee on Integrity chair of the IEEE CS publications board, an associate editor in chief of IT Professional, cochair of the IEEE Reliability Society's Technical Committee on the Internet of Things, and a founding member of the IEEE Special Technical Committee on Big Data. She was the founding chair of the IEEE CS Special Technical Community on Cloud Computing and the editor in chief of* Transactions on Cloud Computing. *You can read her blogs, "Sensing IoT" and "A Cloud Blog," on Computing Now. Contact her at irena.bojanova@computer.org.*

This article originally appeared in IT Professional, *vol. 18, no. 6, 2016.*

The Outcomes of Cybersecurity Competitions and Implications for Underrepresented Populations

Portia Pusey | Portia Pusey, LLC
Mark Gondree | Sonoma State University
Zachary Peterson | California State University, San Luis Obispo



> When I was going through [the Collegiate Cyber Defense Competition (CCDC)] I kept thinking, “Is this really what it’s like?” I’ve been working for two years now (so clearly I don’t know everything about IT or security), but I can say what I learned training for, and competing in, CCDC has helped me more in the real world than 90 percent of the stuff I learned in the classroom. —CCDC participant[1]

Almost every cybersecurity competition organizer could share anecdotes similar to the one above. These types of statements excite employers while making cybersecurity program chairs cringe. But are these positive anecdotes enough to prompt changes to curricula and the integration of competitions into courses? What research has been conducted to unbundle the outcomes of competitions? What evidence do we have to support claims of competition advocates? And can the criticisms be validated?

In 2010, the US Department of Homeland Security Science and Technology Directorate awarded a contract to the US Cyber Challenge to develop a methodology for classifying cybersecurity challenges, games, and competitions. The project reflected the value of and need for an evidence-based approach to understanding the design of cybercompetitions. The results of this exploratory study revealed that little work to date has methodically considered

- the challenges included in a competition, including which vulnerabilities, attack tactics, techniques and protocols, and remediation tasks are simulated during competition;
- the competencies required to perform well in each challenge;
- to what degree competition scores accurately reflect the difficulty of task performance;
- how to align or adjust competition difficulty to student competency levels to ensure participants benefit educationally and build self-efficacy as they master challenges; and
- the effectiveness of competitions in engaging students in cybersecurity—first as a game or simulation, and later as a profession.

In 2013, the Cybersecurity Competition Federation (CCF) was established with NSF support as an association of academic, industry, and government organizations with a common interest in supporting cybersecurity competitions and the competitors they serve. This federation communicates with and promotes cybersecurity competitions to increase awareness, provide guidance on ethical standards, build a common understanding of diverse competition tasks, support those who oversee activities and competitions, and create a developmental pathway using activities that aid the growth of cybersecurity skills. During the three-year grant period, CCF members conducted research to understand the players and outcomes of cybersecurity competitions to identify the needs of competition stakeholders.

Here, we reflect on cybersecurity competitions, drawing primarily from CCF workshops, literature reviews, and reported outcomes of similar STEM (science, technology, engineering, and mathematics) competitions. In particular, we consider those studies relevant to gifted students, females, and low-income and high-risk groups.

Learning Outcomes

Anecdotal evidence, such as the rapid increase in the number and diversity of competitions, shows that students believe competitions can be fun. And there's further complementary evidence that competitions can motivate students to learn. Whether to fulfill formal learning or personal development goals, players might actively connect competition experiences to practice techniques or apply the knowledge they've acquired. Learning outcomes, however, are implicit even when players appear primarily motivated by fun: as they're exposed to different challenges, players expand their ability to apply what they know to solve new problems.

In some STEM programs, competitions are used to measure student growth or as capstone projects. Some instructors use competitions formatively to identify individual students' gaps in knowledge and skills. One educator reported the metacognitive possibilities of competitions: as students work in teams, they're asked to provide one another feedback as well as reflect on their own abilities.[2]

Competitions offer problem-based learning in authentic situations and represent a student-centered approach to knowledge development. A working group on student motivation reported increased and active participation in a postchallenge discussion of solutions.[3] Increases in knowledge and skill attributed to participation in competitions have also been reported.[4] Competitions that are modeled on standardized tests have been used to raise student scores on college entrance exams.[5] Furthermore, there's evidence that team-based competitions support the development of "soft skills" such as teamwork, critical thinking, and communication.[3]

Competitions can also enable differentiated learning and enriched experiences for students with diverse skill levels. One programming competition reported that novices were inspired to apply their learning and improve their projects, while advanced students were incentivized with projects that challenged their abilities. Some competitors, however, report that their educational curriculum doesn't prepare them for competitions.[1]

One plausible explanation for these accounts is the possible disconnect between formal instructional content and competitions; however, multiple other factors are almost certainly involved. Training for cybersecurity competitions might be subject to the same knowledge transfer challenges experienced in physical education: when training is limited to isolated, repetitive practice of techniques, players have difficulty applying those techniques during actual game play. Physical education researchers recommend teaching modified versions of games to situate practice in an authentic framework.[6] This might also contribute to better transfer of formal learning to workplace situations.

Career Preparation Outcomes

Several researchers conclude that competitions build awareness and interest in STEM fields by simulating professional work experiences or using directly transferrable skills,[7] and that students participate in extracurricular activities to build a workforce-ready skill set and resume.[8] In a study of the Science Olympiad—a team competition in which K–12 students compete in events pertaining to various scientific disciplines—76 percent of alumni stated that participation contributed to their professional accomplishments.[9]

Regular participation in extracurricular experiences is correlated with employment and higher pay.[10] Alexander Astin asserted that growth in knowledge and skill is expected because students chose social and extracurricular experiences connected to education.[11] Furthermore, there's evidence that when players choose competitions aligned to career skill sets, they're indicating their active engagement in a profession.[12] However, the larger body of literature on competitions, including cybersecurity competitions, doesn't support the idea that competitions attract and retain diverse populations not already engaged with the subject area.

Diversity Outcomes

Competitions, by nature, rank and filter players. Unintentionally, this can start at the grade-school level, where students might be effectively excluded from competing because they lack access to resources such as sufficient computers and educators with subject-specific training. In some STEM contest designs, only one student advances from each school.[5,9] Diversity is a demonstrated limitation of the Science Olympiads: competitors tend to be male, Caucasian, third-generation Americans with a high socioeconomic status.[9] Furthermore, some school programs prioritize gifted students to improve their competition standing.[7] In contrast, cybersecurity workforce development experts are currently calling to advance the knowledge and skills of those groups underrepresented in the field.[13] Building awareness and engaging students underrepresented in cybersecurity careers support the goals of producing more trained workers to address the deficit in the national workforce pipeline and of increasing the field's overall quality.

The question remains: Once we've built student awareness and interest, how do we support their success in competitions? Participation in extracurricular activities already predicts interest; however, are there factors that predict winning or top ranking? Although a study of Science Olympiad alumni didn't find that age, race, or grade level correlated with finishing in the top ranks, it did identify three significant indicators: type of school, number of previous competitions attended, and number of science courses completed.[14]

Because competition experience and content knowledge are critical factors for successful outcomes, it's important to provide participation opportunities to diverse populations. Indeed, very different social supports and academic interventions might be appropriate when trying to invest in diversity and serve populations underrepresented in the field, including women and students of low socioeconomic status.

Top Performers and Gifted Students

The National Science Board reports that some of America's most talented youth aren't being identified and developed—so we're losing many who have the potential to be the next generation of STEM innovators.[15] Gifted students are typically curious and excellent problem solvers who demonstrate persistence when confronted with a challenge. At the same time, mathematically gifted students can disengage from formal math instruction early on because elementary school educators can't address these students' intuitive understanding of algorithms.[9] Students labeled as gifted might also avoid the pressure of competing against other gifted students because they're discouraged when they discover that they're "not the best."[5] But, ultimately, competitions are one way to educate gifted students: a study on math, chemistry, and physics Olympiad alumni concluded that such competitions effectively advanced their STEM talents.[9]

Low Socioeconomic Status, High Risk

"Students learn by becoming involved."[11] On college campuses, however, first-generation college students aren't likely to join clubs or organizations—despite strong evidence that such involvement is associated with positive outcomes for this population.[8] Students who were involved in clubs during high school or who live or work on campus are more likely to participate in clubs during college. Faculty involvement can also increase student participation.[8] Research suggests that supportive relationships and youth programs let high-risk students overcome obstacles to academic success.[16] Cybersecurity clubs and competitions can succeed in broadening diversity in the workforce pipeline only if recruitment and outreach include long-term interventions such as supportive relationships and early involvement with campus faculty and students.

Gender

Women make up only 11 percent of the information security workforce.[17] A case study investigating the Israeli National Computer Science (CS) Olympiad reported 15 percent female participation in early rounds of the competition, but despite targeted recruitment and participation in advanced training, no woman has ever reached the final.[18] Such attrition is especially startling in light of the following trends: women are more likely than men to enroll and graduate from college and to participate in nonathletic extracurricular activities, and just as likely to use technology such as computers, tablets, and smartphones.[19] Adding to the problem's complexity, it's been reported that almost 50 percent of the middle school students in technology-related classes in the US are female, a number that drops to only 17.7 percent by high school. Therefore, supporting gender equity in competitions requires addressing a larger systemic problem that starts before or during middle school. It's been posited that women don't see the social benefit of a perceived solitary occupation.[20] Others theorize that women experience low self-confidence due to lack of experience or role models.[16] Successful strategies to help engage more women in cybersecurity competitions will involve providing girls with learning experiences and extracurricular activities that build self-efficacy and career engagement before they leave middle school.

Design Considerations

Current cybersecurity competitions claim to offer experiences ranging from novice to expert. Players can find competitions that focus on almost any cybersecurity field: offense, defense, cryptography, forensics, reversing, programming, and any combinations of these. Some competitions are designed for fun or prizes, others for recruitment and identification of talent, and still others for reputation building. The (unadvertised) challenge for players is to find competitions that align with their interests, capabilities, and goals. Existing literature documents several design considerations that would support engagement in competitions and be useful for developing the skills required for the next level of competition. For example, novice coaches and students have frequent questions and require additional support.

One programming-competition developer suggests that organizers give participants the challenge packets two weeks before the competition. This lets participants determine whether they have the adequate skills and interested team members.[21] It's also been suggested that novice competitors replicate best practice in realistic simulations. Several competitions have been designed to help students apply the thoughtful process of planning and implementing security while maintaining the efficiency of network services. This realistic representation is thought to prepare competitors to meet their future employers' needs; however, it might be too complex for novice players.

Novice players also require careful alignment of challenge difficulty to their existing competency. Game balance is achieved when a competition doesn't exceed the players' capabilities. The National Cyber League has developed an innovative approach to providing a competition for players of all skill levels: before individual and team competitions, a mandatory preseason competition is held during which players are bracketed by score, so novice players compete against other novice players, and so on. This method has resulted in a smaller percentage of dropouts among novice populations.[22]

Identifying a player's entry-level competency might be key to successful outcomes in cybersecurity competitions. Karen Cooper found that simulation systems led to engagement only when the participant's skill level was sufficiently high.[23] This finding is corroborated by a small exploratory study that found that competitions might be disengaging to novice learners.[24] Thus, competitions might be effective only for students with existing skill sets that closely match competition requirements. CCF research into competition outcomes determined that competitions used in education require special considerations. Frances Karnes and Tracy Riley list criteria that educators might consider when selecting competitions for their students.[7] In particular, if competitions are to be used in an educational setting, the activities must align with official curriculum. Competitions should be designed with the outcomes for each activity clearly stated. This will help teachers justify inclusion of the competition. Clearly stated objectives also help teachers choose activities that are relevant and interesting to their students.

There's an opportunity to build underrepresented students' self-efficacy by incorporating cybersecurity competitions into the standard K–12 curriculum.

Limitations

Up to this point, we've discussed the promise of competitions. They reward accomplishments in STEM fields and are a tangible expression of STEM's importance and value. Increased program enrollment has also been reported as individuals and teams win competitions.[4] However, the most probable explanation for increased enrollment is the likelihood that competition-related extracurricular programs attract students who are already engaged with the STEM fields and likely to enroll in STEM programs in college.

What's more, the competition literature is filled with unsupported claims of engagement and motivation for learning in classrooms. Anecdotal claims might be connected to any "break from their usual routine"[25] rather than to the competition itself.[26] Further research is required because some case studies of immersive educational simulations support the view that hands-on activities engage the participant and, in so doing, facilitate situational learning and transfer of skills to the real world.[27] Figure 1 lists future research directions for improving the design of cybersecurity competitions.

Although there's been some research on the outcomes and efforts to support engagement of underrepresented populations in cybersecurity competitions, much work remains. For example, most training for cybersecurity competitions occurs through extracurricular activities; so, there's an opportunity to build self-efficacy among underrepresented students by incorporating competitions or challenges into the standard K–12 curriculum by providing hands-on tutorials that let students learn independently or in teams using any Internet-capable computer. We must continue to fund and conduct research that determines cybersecurity competitions' effect on students' awareness

Factors studied in cybersecurity competitions:

- lack of opportunities for novices,
- high attrition, and
- lack of alignment to curricular outcomes.

Unstudied factors, suggested by STEM studies:

- good correlation to professional success,
- rankings' effect on creating incentives to promote or advantage gifted populations, and
- self-selection for participation by second- or third-generation college students.

Implications for future efforts in designing and assessing competitions:

- alignment of winning and scoring with another intended, measurable outcome;
- programs focused on establishing mentorship networks and building self-efficacy;
- programs serving students before or during middle school;
- studies directly paralleling those from STEM competitions, for example, alumni studies, to see whether lessons learned, in fact, translate to cybersecurity competitions; and
- studies validating claims about engagement and learning.

Figure 1. The factors listed can limit cybersecurity competitions' effectiveness in STEM (science, technology, engineering, and mathematics) outreach and are thus potential areas for future research.

of cybersecurity careers and their ability to build confidence and self-efficacy as well as research to establish a developmental pathway of cybersecurity-based activities that support skill growth. ■

Acknowledgments

This material is based on work supported by the NSF (grants 1343536, DUE-1140561, DUE-141931) for the National Cybersecurity Sports Federation and the TableTop Security project (grant 1419318); the USENIX Workshop on Advances in Security Education; and the USENIX Summit on Gaming, Games and Gamification in Security Education. Zachary Peterson's contributions to this work were partially supported by a US–UK Cyber Security Fulbright Scholarship.

References

1. P. Pusey, C. O'Brien, and L. Lightner, "National CyberWatch Center Preparing for the Collegiate Cyber Defense Competition (CCDC): A Guide for New Teams and Recommendations for Experienced Players," 2014; scout.wisc.edu/cyberwatch/downloads/62/NCC_Press_How_To_Prepare_For_the_CCDC.pdf.
2. A. Conklin, "The Use of a Collegiate Cyber Defense Competition in Information Security Education," *Proc. 2nd Ann. Conf. Information Security Curriculum Development* (InfoSecCD 05), 2005, pp. 16–18.
3. J. Carter et al., "ITiCSE 2011 Working Group Report Motivating All Our Students," *Proc. 16th Ann. Conf. Innovations and Technology in Computer Science Education* (ITiCSE 11), 2011, pp. 5–8.
4. J. Rursch, A. Luse, and D. Jacobson, "IT-Adventures: A Program to Spark IT Interest in High School Students Using Inquiry-Based Learning with Cyber Defense, Game Design, and Robotics," *IEEE Trans. Education*, vol. 53, no. 1, 2010, pp. 71–79.
5. A. Trotter, "Competing for Competence," *Education Week*, vol. 27, no. 30, 2008, pp. 36–38.
6. P. Werner, R. Thorpe, and D. Bunker, "Teaching Games for Understanding: Evolution of a Model," *J. Physical Education, Recreation & Dance*, vol. 67, no. 1, 1996, pp. 28–33.
7. F.A. Karnes and T.L. Riley, "Developing an Early Passion for Science through Competitions," *Gifted Child Today*, vol. 22, no. 3, 1999, pp. 34–36.
8. K.F. Case, "A Gendered Perspective on Student Involvement in Collegiate Clubs and Organizations in Christian Higher Education," *Christian Higher Education*, vol. 10, nos. 3–4, 2011, pp. 166–195.
9. J.R. Campbell and H.J. Walberg, "Olympiad Studies: Competitions Provide Alternatives to Developing Talents that Serve National Interests," *Roeper Rev.*, vol. 33, no. 1, 2010, pp. 8–17.
10. D.D. Albrecht, D.S. Carpenter, and S.A. Sivo, "The Effect of College Activities and Grades on Job Placement Potential," *NASPA J.*, vol. 31, no. 4, 1994, pp. 290–297.
11. A.W. Astin, "Student Involvement: A Developmental Theory for Higher Education," *J. College Student Personnel*, vol. 25, no. 4, 1984, pp. 297–308.
12. M. Prenzel, "The Selective Persistence of Interest," *The Role of Interest in Learning and Development*, K.A. Renninger, S. Hidi, and A. Krapp, eds., Lawrence Erlbaum Associates, 1992, pp. 71–98.
13. "National Initiative for Cyber-Security Education: Strategic Plan," National Initiative for CyberSecurity Education, 2012; csrc.nist.gov/nice/documents/nicestratplan/nice-strategic-plan_sep2012.pdf.
14. W. Baird, "Correlates of Student Performance in the Science Olympiad: The Test of Integrated Process Skills and Other Variables," *Proc. Ann. Meeting National Assoc. Research in Science Teaching*, 1989; files.eric.ed.gov/fulltext/ED305248.pdf.
15. *Preparing the Next Generation of STEM Innovators: Identifying and Developing Our Nation's Human*

Capital, report NSB 10-33, National Science Foundation, 2010.

16. T.G. Zimmerman et al., "Why Latino High School Students Select Computer Science as a Major: Analysis of a Success Story," *ACM Trans. Computing Education*, vol. 11, no. 2, 2011, p. 10.
17. *Agents of Change: Women in the Information Security Profession: The (ISC)^2 Global Information Security Workforce Subreport*, Frost & Sullivan, 2013; www.isc2cares.org/uploadedFiles/wwwisc2caresorg/Content/Women-in-the-Information-Security-Profession-GISWS-Subreport.pdf.
18. O. Sagy and O. Hazzan, "Diversity in Excellence Fostering Programs: The Case of the Informatics Olympiad," *J. Computers in Mathematics and Science Teaching*, vol. 26, no. 3, 2007, pp. 233–253.
19. C.R. Mitts, "Gender Preferences in Technology Student Association Competitions," *J. Technology Education*, vol. 19, no. 2, 2008, pp. 45–59.
20. P. Doerschuk, J. Liu, and J. Mann, "Pilot Summer Camps in Computing for Middle School Girls: From Organization through Assessment," *ACM SIGCSE Bull.*, vol. 39, no. 3, 2007, pp. 4–8.
21. L. Sherrell and L. McCauley, "A Programming Competition for High School Students Emphasizing Process," *Proc. 2nd Ann. Conf. Mid-south College Computing* (MSCCC 04), 2004, pp. 173–182.
22. C. O'Brien, P. Pusey, and J. Jones, "Competition as Curriculum: Why Competitions Will Work in Your Field," *Proc. High Impact Technology Exchange Conf.: Educating America's Technical Workforce* (HI-TEC 14), 2014.
23. K. Cooper, "Go with the Flow: Engagement and Learning in Second Life," *Proc. Spring Simulation Multi Conf.* (SpringSim 10), 2010; doi:10.1145/1878537.1878578.
24. D.H. Tobey, P. Pusey, and D. Burley, "Engaging Learners in Cybersecurity Careers: Lessons from the Launch of the National Cyber League," *ACM Inroads*, vol. 5, no. 1, 2014, pp. 53–56.
25. A. Rosenbloom, "Running a Programming Contest in an Introductory Computer Science Course," *Proc. 14th Ann. ACM SIGCSE Conf. Innovation and Technology in Computer Science Education* (ITiCSE 09), 2009, p. 347.
26. R.E. Clark, "Reconsidering Research on Learning from Media," *Rev. Educational Research*, vol. 53, no. 4, 1983, pp. 445–459.
27. C. Dede, "Immersive Interfaces for Engagement and Learning," *Science*, vol. 323, no. 5910, 2009, pp. 66–69.

Portia Pusey is principal at Portia Pusey, LLC. Contact her at edrportia@gmail.com.

Mark Gondree is an assistant professor of computer science at Sonoma State University. Contact him at gondree@sonoma.edu.

Zachary Peterson is an associate professor of computer science at California State University, San Luis Obispo. Contact him at znjp@calpoly.edu.

This article originally appeared in EEE Security & Privacy, *vol. 14, no. 6, 2016.*

Cloud Manufacturing: Security, Privacy, and Forensic Concerns

Christian Esposito and Aniello Castiglione
University of Salerno

Ben Martini
University of South Australia

Kim-Kwang Raymond Choo
University of Texas at San Antonio

OVER THE LAST FEW DECADES, THE WAY MANUFACTURING ENTERPRISES HAVE BEEN MANAGED HAS UNDERGONE A RADICAL RETHINKING.[1] This has led to the so-called Industry 4.0, or the fourth industrial revolution, and traditional management models have been progressively abandoned. A concrete example of this trend is the European Factories of the Future Research Association, a public-private partnership under Horizon 2020.[2] EFFRA has produced a roadmap to pave the way for introducing innovation-driven transformations within the European manufacturing sector.

Such a tremendous boost for innovation in manufacturing arises from the current economic environment, which is extremely volatile and globalized. Enterprises need to rapidly respond to changing or uncertain market demands, provide customized products and services, and compete at the international level by targeting multiple potential markets around the world. Enterprises are deemed successful if they can provide a wide variety of high-quality products while keeping manufacturing and distribution costs low to meet customer expectations and needs. Moreover, the contemporary need to target multiple markets in different countries requires enterprises to expand their production capability by setting up multiple manufacturing sites around the world.

The *networked manufacturing* framework,[3] illustrated in Figure 1, interconnects the strategic centers of an enterprise, enabling it to operate at a worldwide scale. This is different from a logistic network, where products are exchanged to lower production costs. The networked manufacturing framework envisions the exchange of products, associated services, and knowledge to improve productivity, flexibility, and competitiveness. Networked manufacturing is a concrete realization of distributed manufacturing where a network is used to integrate production and shipping facilities, with the headquarters playing the role of centralized manager for the overall network by monitoring and adjusting the day-to-day contingencies and activities.

Models such as networked manufacturing started as intrafirm organizational models to address the globalization needs of enterprises, but later evolved into a collaborative approach between firms. The issues and challenges of the current economic environment are making it more difficult to run small and medium enterprises (SMEs), since they don't have the skills and resources required to compete against larger enterprises. Therefore, SMEs are joining efforts and capabilities to overcome their limitations through collaboration, which can be short term or more stable and durable.

The collaborative networked manufacturing model, depicted in Figure 2, has paved the way for more advanced organizational structures, such as

2469-7087/17/$33.00 © 2017 IEEE

virtual enterprises or virtual organizations, which allow businesses or public services to join forces to better respond to business opportunities and needs. Both intrafirm and interfirm collaborative networks are giving rise to novel forms of organizations and establishing a more pervasive role for information sharing within the current manufacturing practice.[4] In fact, most of these collaborative approaches are based on rich and efficient information sharing, which supports proper scheduling and monitoring of facility costs, performance and flexibility, decision making, and management of the network complexity in terms of integrated firms and facilities.

The Advent of Cloud Manufacturing

Smart manufacturing increases competitiveness and efficiency through interconnection and cooperation among companies or among resources within a single company. Recent advances in information and communication technologies (ICT) that support collaboration and cooperation among organizations

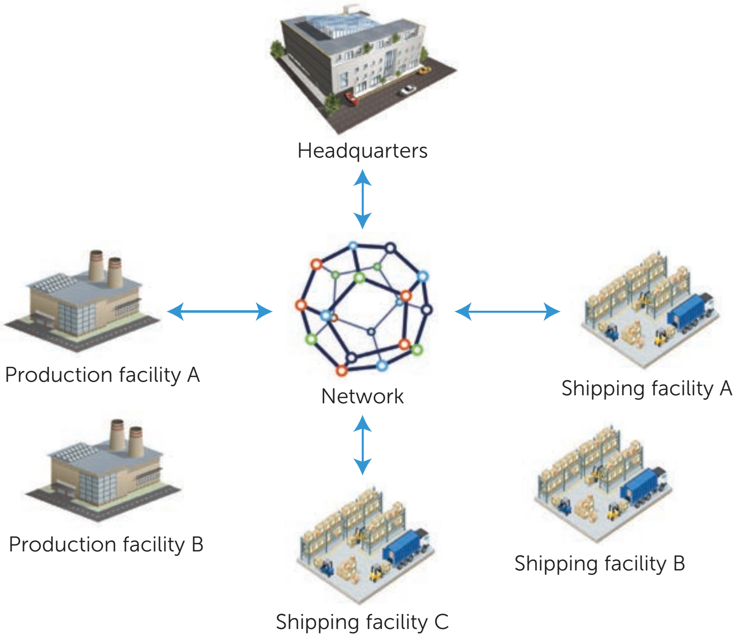


FIGURE 1. Schematic representation of networked manufacturing.

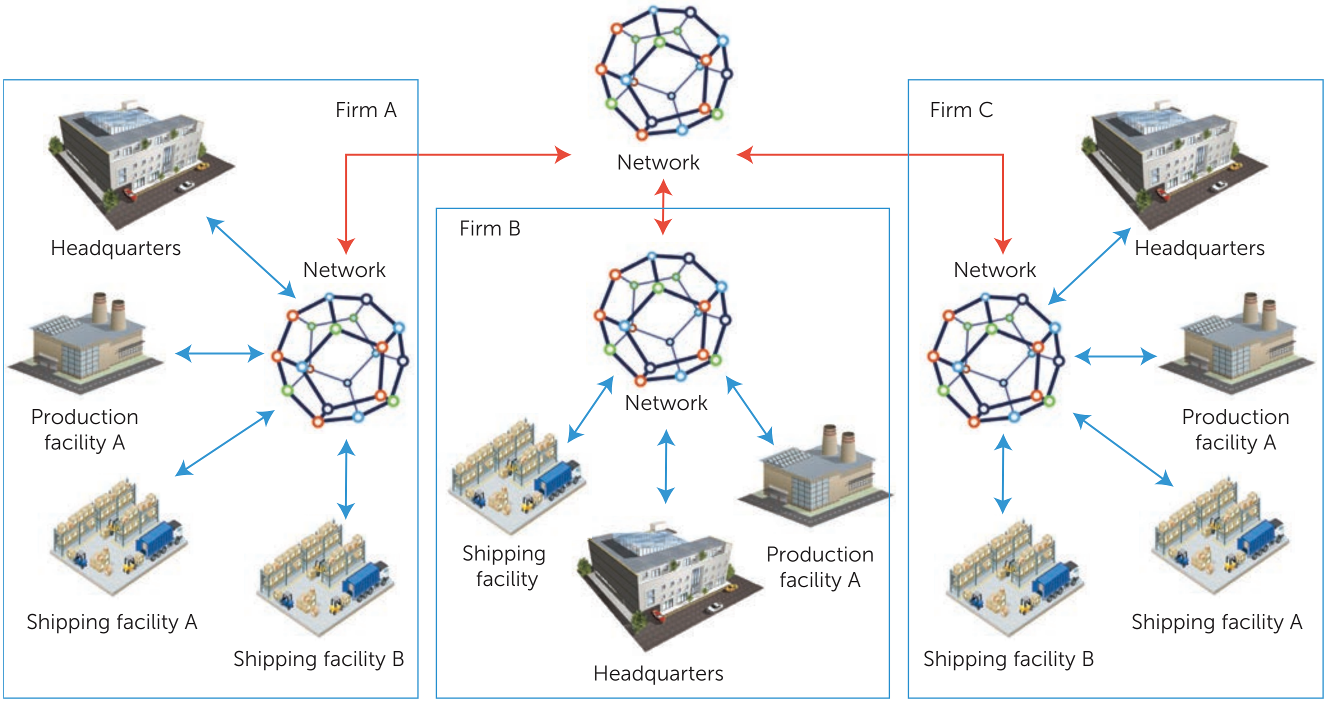


FIGURE 2. Schematic representation of collaborative networked manufacturing.

beyond mere information exchange, to the realization of knowledge and service sharing, have made this vision possible. Open source/Web-based applications are considered key enablers of integrated enterprise practices and strategies for the manufacturing industry due to their flexibility, interoperability, and proliferation.[5] However, the increasing complexity of managing the software needed to deal with these collaborative schemes makes computing extremely expensive for an enterprise, particularly SMEs. Hence, the proliferation of collaborative networked manufacturing solutions within the manufacturing sector is slowing.

The emergence of cloud computing within the business environment offers a solution.[6] Cost is the main driver of enterprises' adoption of cloud computing. The cloud's pay-per-use cost model lets enterprises reduce capital investments in information technology, leading to significant cost savings. However, cloud computing does more than provide cost-effective computing; it also provides for the flexible provisioning of ICT resources, with its elasticity allowing for rapid scaling to the dynamic and ever-changing needs of enterprises.

Two approaches for adopting cloud computing within the manufacturing domain have emerged as most promising:

- the naïve and direct use of cloud platforms as data sharing and storage enablers to support collaboration schemes, as exemplified in the collaborative networked manufacturing model; and
- cloud manufacturing, where distributed resources within the networked manufacturing framework participating in a manufacturing business process are modelled and encapsulated as cloud services and managed in a centralized manner.[7]

Such a solution is increasingly being applied within industrial practice. An IBM industry survey revealed that two-thirds of mid-sized companies have already implemented or are about to migrate to a cloud-based storage model.

Figure 3 shows a cloud manufacturing application model, which can be realized using a layered service-oriented architecture. At the lowest level is a set of manufacturing resources (the physical facilities or capabilities) within a firm or across multiple firms, required to move the product through the development lifecycle. The next layer contains cloud services that virtualize, encapsulate, and identify the underlying manufacturing resources, which are responsible for executing manufacturing tasks while ensuring high production quality and reliability. The service layer encompasses cloud services built on top of the virtualized manufacturing resources to implement remote monitoring, scheduling, and control of manufacturing resources. The last level is the application layer, which presents a set of services acting as interfaces for users to the cloud manufacturing solutions. The provided operations allow designers and administrators to model manufacturing processes, perform these processes by properly integrating and composing virtual resources, and monitor a running manufacturing process by visualizing some measures of merit.

The radical rethinking of the manufacturing industry from the traditional production-oriented approach to the service-oriented one envisioned by the networked manufacturing framework, collaborative networked manufacturing, and cloud manufacturing faces some obstacles in the current industry environment.[8] Foremost are the safety and security issues such collaborative schemes present. The networks used to support collaborations and cooperation convey business-critical information, while the virtualization and service orientation of manufacturing resources make enterprises vulnerable to a new series of attacks not seen in traditional manufacturing approaches.

Today, security is a key concern when using cloud computing in mission-critical scenarios, including the manufacturing domain. A cloud manufacturing solution could be compromised, and critical data could be stolen or altered by amateur attackers. Experts, perhaps hired by competitors, could also compromise a cloud system, significantly affecting an organization's productivity and reputation. Therefore, equipping cloud manufacturing solutions with proper security management mechanisms and policies is critical to avoid possible threats to both the solution and consumers.

Secure Cloud Manufacturing

As a recent Cloud Security Alliance (CSA) report noted, data breaches are among the most frequent se-

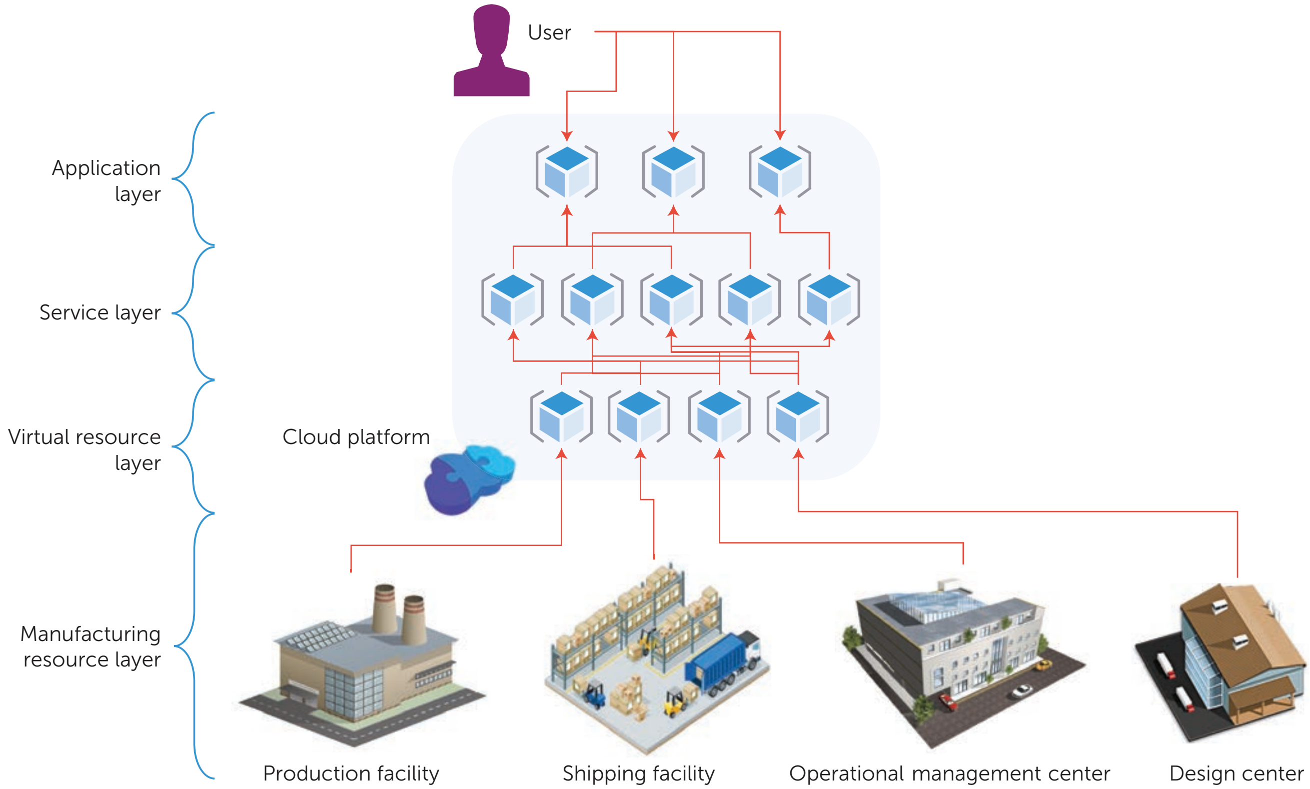


FIGURE 3. Layered architecture of cloud manufacturing.

curity compromises[9] and have proved to have tremendous consequences from both legal and economic viewpoints. A cloud data breach is an unauthorized or illegal access to data hosted within the cloud, in terms of both retrieving and modifying data. This is a particular concern within the context of cloud manufacturing, where a data breach can result in the loss of sensitive corporate information, such as trade secrets or contract details, and consequently negatively affect the company's reputation. Encryption, the most common solution to prevent data breaches,[10,11] is offered in several cloud platforms.

However, encrypting data that's outsourced to the cloud doesn't solve the problem. Within the context of cloud manufacturing, sources of data breaches are typically within the company rather than outside it. In fact, a malicious insider, such as a current or past employer, a system administrator, a contractor, or a business partner, might be the culprit, as the CSA report indicates.[9] Some cloud platforms facilitate the use of encryption keys controlled by their customers. This isn't effective in case of malicious insiders, who might possess the correct keys for the decryption. Manufacturers using cloud services therefore need to also adopt proper key management best practices that go beyond the technical aspects to consider organizational and social perspectives for security assurance.[12] Such best practices are typically a set of reasonable guidelines and considerations for an effective strategy over all seven phases of a key management process, as Figure 4 illustrates.

- A key should be produced using a cryptographic module with at least FIPS 140-2[13] compliance for memory, strength, and secrecy requirements.
- An appropriate key distribution approach must be used to distribute the key to all authorized users to ensure its secrecy.[14]
- Users can choose to persistently store keys in key stores that provide secrecy guarantees.

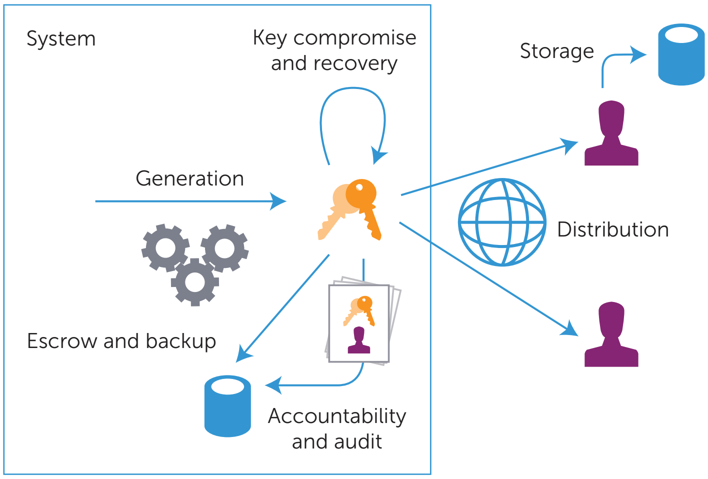


FIGURE 4. Phases of key management.

- Because keys can be lost, systems with data at rest for long periods need to have a key recovery plan.
- To prevent key compromises or reduce their impact when detected, a system should document which entity received or had control of a certain key.
- When a key is known to have been compromised, it must be revoked and new keys generated and distributed to authorized entities.

A strategy for managing cryptographic keys is only the first defense against data breaches. Because data breaches are almost inevitable, systems must be equipped with a means to identify and document them and to notify relevant personnel. Specifically, it's necessary to have means to determine that data has been read or changed by an unauthorized entity, to inform the data owner that a breach has occurred, and to collect and store, in a forensically sound manner, all the information related to the breach and the suspected culprit. Recent laws and regulations for data protection have detailed how to notify and document data breaches, highlighting the importance of this concern. The ePrivacy Directive (2002/58/EC), for example, introduced a European data breach notification requirement for the electronic communication sector.[15]

An effective cloud manufacturing data loss solution should support the four stages of prevention, identification, notification, and documentation of data breaches. Such a solution could be deployed as software as a service (SaaS), as Figure 5 illustrates, to be easily integrated into current operational processes in the manufacturing domain (in fact, it can be easily extended to other domains with similar requirements). Specifically, such a solution should be equipped with a module for breach notification and one for documentation according to relevant standards and regulations. To facilitate seamless and simple notifications, we envision the use of a secure publish/subscribe service—that is, a middleware solution for the asynchronous and confidential exchange of breach information with interested parties.[16]

Criminal data breaches would be of particular interest to law enforcement, and specifically digital forensic practitioners. This issue is important to manufacturing companies, since data breaches can ruin their reputation and market opportunities and give their competitors an advantage. Companies must be able to defend their copyrights in court and successfully prosecute the culprits behind data breaches. Companies could use digital forensic techniques to ensure that evidence collected as part of a data breach event remains forensically sound (that is, suitable to be upheld as original evidence in court). This process starts with initial preservation (that is, collection) and continues through transmission to law enforcement, and ultimately presentation in court. The large body of digital forensic literature can assist in the development of this part of the process.[17,18]

Another module should be devoted to the application of an effective key management strategy, according to given standards and regulations, such as the one issued by the US National Institute for Standards and Technology.[19] Key management is a serious concern in the manufacturing domain, since it's the main factor allowing data breaches. As previously stated, malicious insiders could obtain valuable documents and trade secrets related to a company's products and manufacturing processes, share them with competitors, or use them to start their own business. Preventing such breaches requires a proper key management system to record which employees hold certain keys and revoke the keys when employees don't need them. Key management should be stringent to avoid the possibility of violations, but not so strict that employees can't do their jobs effectively.

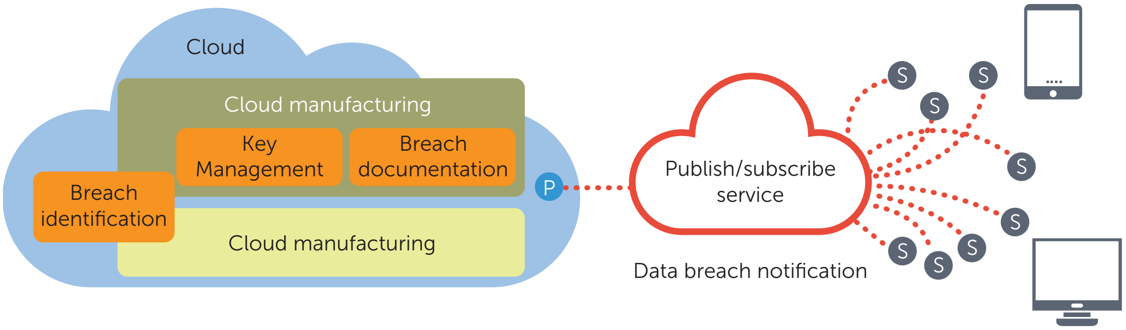


FIGURE 5. Software-as-a-service (SaaS) solution for defending against data breaches in cloud manufacturing.

Despite all of the preventive measures put in place by a company, a data breach can still occur. A manufacturing company must be able to promptly detect a data breach to prevent malicious insiders and competitors from further exploiting the vulnerability to obtain company documents and secrets. The last module in the SaaS solution we envision is responsible for identifying such breaches. The breach identification module will monitor the data exchanged and stored within clouds when a given manufacturing process is performed, checking the correct flow of data within the overall infrastructure.

Breach identification remains an open research issue, and lacks a substantial body of literature. One possible solution is to use digital watermarking and other steganography-based approaches on the data held by the cloud manufacturing solution. The principle is to include data that can be used to detect possible unauthorized modifications or access resulting from a data breach.

FUTURE WORK IN THIS AREA INCLUDES IMPLEMENTING A PROTOTYPE OF OUR SOLUTION IN A REAL-WORLD ENVIRONMENT WITH THE AIMS OF EVALUATING OUR SOLUTION, AND REFINING IT IF NECESSARY. Other possible future research directions include investigating reliability and fault-tolerance issues in cloud manufacturing, the relationship or influence reliability and fault-tolerance issues have on security issues, and the possibility of a holistic approach for these two complementary aspects. ●●●

References

1. Y. Koren, *The Global Manufacturing Revolution: Product-Process-Business Integration and Reconfigurable Systems*, John Wiley & Sons, 2010.
2. European Factories of the Future Research Association, *Factories of the Future: Multi-Annual Roadmap for the Contractual PPP under Horizon* 2020, report, 2013; www.effra.eu/attachments/article/129/Factories%20of%20the%20Future%202020%20Roadmap.pdf.
3. B. Montreuil, J.-M. Frayret, and S. D'Amours, "A Strategic Framework for Networked Manufacturing," *Computers in Industry*, vol. 42, nos. 2–3, 2000, pp. 299–317.
4. S. D'Amours et al., "Networked Manufacturing: The Impact of Information Sharing," *Int'l J. Production Economics*, vol. 58, no. 1, 1999, pp. 63–79.
5. L.M. Camarinha-Matos et al., "Collaborative Networked Organizations: Concepts and Practice in Manufacturing Enterprises," *Computers & Industrial Eng.*, vol. 57, no. 1, 2009, pp. 46–60.
6. S. Mareston et al., "Cloud Computing: The Business Perspective," *Decision Support Systems*, vol. 51, no. 1, 2011, pp. 176–189.
7. X. Xu, "From Cloud Computing to Cloud Manufacturing," *Robotics and Computer-Integrated Manufacturing*, vol. 28, no. 1, 2012, pp. 75–86.
8. D. Wu et al., "Cloud Manufacturing: Strategic Vision and State-of-the-Art," *J. Manufacturing Systems*, vol. 32, no. 4, 2013, pp. 564–579.
9. Cloud Security Alliance, *The Treacherous 12:*

Cloud Computing Top Threats in 2016, tech. report, Top Threats Working Group, Feb. 2016; https://downloads.cloudsecurityalliance.org/assets/research/top-threats/Treacherous-12_Cloud-Computing_Top-Threats.pdf.

10. L. Townsend, "How to Prevent a Data Breach in the Cloud," blog, Townsend Security, 2016; http://info.townsendsecurity.com/bid/63294/How-to-Prevent-a-Data-Breach-in-the-Cloud.
11. B. Grobauer, T. Walloschek, and E. Stocker, "Understanding Cloud Computing Vulnerabilities," *IEEE Security & Privacy*, vol. 9, no. 2, 2011, pp. 50–57.
12. Open Web Application Security Project, "Key Management Cheat Sheet," 2016; www.owasp.org/index.php/Key_Management_Cheat_Sheet.
13. US Nat'l Inst. for Standards and Technology, *Security Requirements for Cryptographic Modules*, Federal Information Processing Standards Publication, May 2011, http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf.
14. K.-K.R. Choo, *Secure Key Establishment*, Advances in Information Security vol. 41, Springer, 2009; http://dx.doi.org/10.1007/978-0-387-87969-7.
15. European Parliament and the Council of the European Union, Directive 2002/58/EC of the European Parliament and of the Council of 12 July 2002 Concerning the Processing of Personal Data and the Protection of Privacy in the Electronic Communications Sector (Directive on Privacy and Electronic Communications), July 2002; http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:32002L0058:en:HTML.
16. C. Esposito and M. Ciampi, "On Security in Publish/Subscribe Services: A Survey," *IEEE Comm. Surveys and Tutorials*, vol. 17, no. 2, 2015, pp. 966–997.
17. B. Martini and K.-K.R. Choo, "An Integrated Conceptual Digital Forensic Framework for Cloud Computing," *Digital Investigation*, vol. 9, no. 2, 2012, pp. 71–80.
18. D. Quick, B. Martini, and K.-K.R. Choo, *Cloud Storage Forensics*, Syngress Publishing/Elsevier, 2013.
19. US Nat'l Inst. for Standards and Technology, *Recommendation for Key Management*, NIST Special Publication 800-57, July 2012; http://dx.doi.org/10.6028/NIST.SP.800-57p1r3.

CHRISTIAN ESPOSITO *is adjunct professor at the University of Naples "Federico II," Italy, and a research fellow at the University of Salerno, Italy. His research interests include information security and reliability, middleware, and distributed systems. Esposito has a PhD in computer engineering from the University of Naples "Federico II." Contact him at christian.esposito@dia.unisa.it.*

ANIELLO CASTIGLIONE *is adjunct professor at the University of Salerno, Italy, and at the University of Naples "Federico II," Italy. His research interests include security, communication networks, information forensics and security, and applied cryptography. Castiglione has a PhD in computer science from the University of Salerno, Italy. He is member of several associations, including IEEE and ACM. Contact him at castiglione@ieee.org.*

BEN MARTINI *is a research fellow at the University of South Australia. His research interests include cybersecurity and digital forensics, focusing on contemporary technologies such as cloud computing and mobile devices. Martini has a PhD in digital forensics from the University of South Australia. Contact him at ben.martini@unisa.edu.au.*

KIM-KWANG RAYMOND CHOO *holds the Cloud Technology Endowed Professorship in the Department of Information Systems and Cyber Security at the University of Texas at San Antonio. His research interests include cyber and information security and digital forensics. Choo has a PhD in information security from Queensland University of Technology, Australia. He's a senior member of IEEE and a Fellow of the Australian Computer Society. Contact him at raymond.choo@fulbrightmail.org.*

This article originally appeared in IEEE Cloud Computing, *vol. 3, no. 4, 2016.*

Evil Offspring – Ransomware and Crypto Technology

Hilarie Orman • *Purple Streak*

Twenty years ago I encrypted a file that I was editing. Ironically, it concerned encryption. The file was on a Unix time-sharing system, and I needed to keep it confidential. Encrypting it meant that I need not rely on the access controls on Unix, which were easily circumvented through bugs in privileged programs. A few days later I returned to edit the file, but I had forgotten the password that unlocked the encryption key. Out of an abundance of caution and foolish faith in my own memory, I had not written it down. I had to start all over to create the file.

My brush with denial of service by encryption didn't suggest a new business venture, but that just shows my lack of imagination. Though I didn't realize it, there were already the rumblings of a lucrative revenue stream enabled by malicious encryption. Now, modern encryption methods have become the basis for monetizing malware. Several varieties of "crypto ransomware" have evolved that take advantage of modern encryption technology. The evil code encrypts all your files, deletes your backups, and asks for a Bitcoin payment in exchange for the decryption key. Hospitals, police departments, small businesses, and ordinary individuals have been faced with the choice of abandoning their data or paying the ransom.

Crypto ransomware is an interesting kind of new crime, one enabled by asymmetric cryptography, block-chaining systems, a large network of botnets, and the fact that no matter how much we wish otherwise, the software that drives our computing devices always has exploitable bugs.

Crypto ransomware is worrisome from a national security standpoint. In the classic treatise *The Art of War*,[1] there's a theme of achieving an advantage through position, preparation, or surprise. But with software technology, it's possible that any advantage can be replicated and turned against an enemy, be it a defender or attacker. Also, consider the section 6 item in that document as advice regarding a zero-day attack:

> *The spot where we intend to fight must not be made known; for then the enemy will have to prepare against a possible attack at several different points; and his forces being thus distributed in many directions, the numbers we shall have to face at any given point will be proportionately few.*

The reality of today's software is that the defenders have all too large an attack surface. Compare this to a recent government report on cybersecurity R&D plans.[2] The report states a goal of achieving advances "to reverse adversaries' asymmetrical advantages" within 3 to 7 years. Crypto ransomware's cleverness might show that such a goal will be very difficult to achieve.

If you have occasion to do forensic analysis or recovery on crypto ransomware, or if you're trying to design countermeasures, it will be useful to know the span of options available to the malware writers and how they might be tripped up or deflected.

History

The first crypto ransomware was probably the infamous AIDS Trojan[3] in 1990. It was distributed on a floppy disk handed out to attendees at an international conference about the AIDS disease, and the software encrypted file names (not the files themselves), and then displayed a demand for payment to a location in Panama. The perpetrator's motivation might have been rooted more in a desire for revenge on the conference organizers than in financial gain, but in any case, the attack was ineffective. The exact reason for this wasn't published, but a program for restoring the file names was quickly distributed.

2469-7087/17/$33.00 © 2017 IEEE

A good guess about how the restoration could have worked illustrates the first principles of successful ransomware: it must be easily reversible, and it must also resist collusion.

The AIDS Trojan probably used the same key for all its encryptions. If someone paid the ransom, then the perpetrator, should he wish to preserve his reputation as a "fair" businessman, could tell the victim what the key was, perhaps by return post. Because the file names were encrypted with a symmetric cipher, it would be easy for the virus software to decrypt the file names when given the key.

But anyone who got the key, either by paying for it or guessing it, could simply tell everyone else, and the scheme would fall apart quickly. I suspect that the software did a very poor job of hiding the key, and that was the basis for the restoration program. It was unnecessary for anyone to pay the ransom.

This early effort didn't kick off a wave of imitators, even though the Internet was making malware virus distribution easier each year. There were a handful of virus programs that used encryption to render a machine useless and demand a ransom, but these used symmetric encryption and were easily undone because they used one key for all encryptions and didn't hide that key very well.

To turn crypto ransomware into a truly dangerous attack, there were two more pieces of technology needed. Asymmetric cryptography was one of them, and although it had been invented two decades earlier and was readily available through Pretty Good Privacy (PGP) software and the GNU Multiple Precision (MP) library, it didn't gain much traction with the malware crowd. This was odd, because in 1996, Adam Young and Moti Yung published a paper describing exactly how to do this.[4] Their method involved generating a unique symmetric encryption key for each infected computer and then encrypting that with a master public key embedded in the virus software. The beauty of their method was that the infected machine didn't need to communicate with the perpetrator until the ransom was paid. At that time, the victim could post the public key encryption of the symmetric key, and the perpetrator could decrypt that and send the symmetric key back to the victim for decryption of the files.

Malware authors didn't pick up on this scheme for about 10 years. Maybe they didn't trust the anonymity or security of the keys, or maybe they were making too much money from other schemes. Or maybe they were wary of collecting payments. Although scams taking advantage of international banking were common, ransomware faced more difficult hurdles to remain hidden. In an ordinary scam, the victims were unlikely to realize their mistake for several days, but with ransomware, the victims would be calling law enforcement immediately, and the bank account would be tracked or shut down quickly. To reliably evade detection, the perpetrators needed anonymous payment. In 1996 there wasn't much in the way of digital cash, but help was just around the corner. Block-chaining and Bit-Coin to the rescue!

How It Works

Cryptography isn't an absolute necessity for ransomware, but it's the only way to get close to an unbreakable denial-of-service extortion attack.

Nonetheless, social engineering and a well-chosen price point can make even non-cryptographic ransomware ("locker ransomware" or "lockerware") an effective tool. Lockerware will divert the computer from its normal operation by getting control of a critical resource, perhaps by encrypting and replacing that resource, and then displaying a seemingly unremovable view of a demand for payment. The demand might appear to come from a law enforcement agency. Some lockerware uses a simple Javascript technique to take control of a browser, again with a ransom demand. If the ransom is paid, the user should receive instructions on how to regain control of his computer or browser.

A particularly insidious way of installing lockerware is to offer a fake antivirus scanning program via a website. The website will pop up a window claiming to have discovered a virus on the visitor's computer screen and will offer a free detection program. The installed software is really malware that will lock up the computer and display an extortion demand. There are many other clever ways of getting users to install software from untrusted sources, but the fake AV trick is the one I think is truest to the ancient story of the Trojan Horse.

If the lockerware ransom amount is low enough, users might pay up rather than spending time searching for information or services to disable the malware. Disabling it might be time-consuming or obscure (like restoring an overwritten master boot record), or even impossible for the general user (as we'll see with Internet of Things devices). Even if the convenience of paying the extortionist seems like an attractive option, victims should be extremely wary of paying it, because there's no guarantee whatsoever that the machine will actually be unlocked.

Several years after the Young and Yung paper, public key ransomware turned up in Russia.[5] There were some fears that the malware had unbreakable cryptography, but the early versions were still primitive things with symmetric ciphers and embedded keys. As with any disruptive technology, it took some years to refine it into a reliable, profitable, worldwide operation. Besides the necessary software engineering skills and an easily usable payment method, businesses need distribution networks, knowledge of optimal price points, revenue-sharing arrangements, and a reliable-yet-anonymous Internet presence. The shadowy figures behind ransomware kept building up their business components, and the industry seemed to reach some kind of fruition a few years ago. Today, most people know about ransomware and probably know someone who was affected by it.

By 2009, crypto ransomware had entered the public key cryptography arena in force, and its use is increasing rapidly. Unlike lockerware, there's no simple way to restore a critical resource and regain normal operation. The computer's files, accounting data, document drafts, contact lists, and so on — all have been transformed into encrypted data and only the encryption key will undo the damage.

From a technology perspective, successful ransomware must meet a handful of critical requirements.

1. Some resource that's valuable to the user must be made unavailable (denial of service).
2. The denial of the resource and the payment instructions must be announced to the user of the afflicted machine in an unavoidable, visible process.
3. The ability to restore the valuable resource must depend on a small amount of data that's available only to the extortionist and can't be inferred or calculated by any other process at reasonable cost.
4. The extortionist must be able to verify payment.
5. The extortionist must be able to accept payment and supply the information for restoring the resource without identifying himself.
6. The restoration process must run on the afflicted computer, it must be simple to use, and the restoration must be reasonably reliable.

Public key cryptography provides the means for achieving requirement, as noted in the Young and Yung paper. However, the only means of getting the strictest sense of "can't be inferred or calculated" would limit the ransomware to a painfully slow public key encryption method. Most ransomware trades off some security for performance, and this gives it the ability to encrypt more of the user's file data before being detected.

Symmetric Keys Only

Apparently the simple way of using symmetric encryption to enable unbreakable ransomware was never used, but it deserves some consideration in the taxonomy of techniques. There are no public keys in this method, and it illustrates the design options open to ransomware developers.

If each instance of the virus used a unique symmetric encryption key for its dirty work, and if it destroyed that key after using it, then file recovery would be nearly impossible. The only problem is that the extortionist must know what that key is in order to release the victim's files. Thus, the victim's machine has to hold some piece of data that that lets the extortionist know which key was used for that victim. Somehow, there must be communication between the extortionist and the victim's machine.

The malware can initiate that communication prior to beginning its work, or it can be done when it finishes encrypting. In the former case, the malware contacts the extortionist and receives a symmetric encryption key and a key identifier. In the latter case, the malware generates a random symmetric key and a random identifier and sends those to the extortionist. In both cases, after encrypting the files, the malware destroys the encryption key but retains the key identifier. If the victim pays the ransom and communicates the identifier to the extortionist, the extortionist will be able to send the corresponding encryption key.

If the victim's machine isn't connected to the Internet, then this attack might fail to get started, or it might fail to leave any way for the victim to recover his data. After the symmetric key is erased, we can only hope that the extortionist actually has the key and the key identifier!

Although this scheme is at the core of all crypto ransomware, as described here it has a serious flaw. Anyone who observes the communication between the malware and the extortioner will be able to see the symmetric key. It might show up in logs of network traffic, either locally or on a network monitor in the communication pathway. However, if the victim has no access to the messages, the method is quite sound.

Embedded Master Public Key

By using public key cryptography, ransomware can avoid the necessity of communicating directly with the extortionist. This is by far the simplest way of implementing ransomware. The method is similar to that in the previous section, but with a crucial difference: the malware has the extortionist's public key embedded in its software.

The malware begins by generating a random key for symmetric encryption. After encrypting the victim's files, the malware uses the embedded public key to encrypt the random ransom key. If the malware leaves no trace of the symmetric key, then the encrypted random key serves the job of the key identifier. After paying the ransom, the victim sends the encrypted key to the extortionist or publishes it in a pre-agreed place. The extortionist will use his private key to unlock the random symmetric key, and he can send it to the victim or publish it in a pre-agreed place.

This method has only two drawbacks. One is that the symmetric key might be visible if a suspicious victim dumps memory while the encryption is active. The other problem is that should the extortionist somehow leak the value of the private key, then all victims could use it to recover their data by decrypting their locally encrypted symmetric key. In fact, one extortioner ended his scheme by publishing the private key.[6] Perhaps these people sometimes experience remorse.

A Unique Public for Each Malware Instance

By adding one roundtrip message, ransomware can avoid the reliance on a single public key. Although most ransomware uses public keys that can't be "broken" in any reasonable computing scenario, still, one public key is only one layer of protection for the extortioner.

If the malware sends a request message to the extortioner's message service, such as a compromised website providing anonymity for the criminals, then the command and control center for the ransomware can send back a freshly computed public key. The malware on a victim's computer will encrypt the symmetric key using the public key. The public key itself serves as the identifier to use when paying the ransom. The extortioner's software will find the matching private key and send it to the victim.

An interesting variant on this method allows the malware to avoid using symmetric encryption. The symmetric methods have a point of vulnerability in that they have to keep the symmetric key in memory for the entire time that the user's files are being encrypted. If the process is interrupted, an examination of memory might reveal the key.

By using public key encryption, the malware will incur a huge time cost penalty. The user might detect that infection before many files are affected. However, the public key encryption methods will yield no useful information about decrypting the files. Only the matching private key, held by the extortioner, can undo the damage.

The Ransom Payment

Bitcoin or other anonymous payment systems protect the extortionists by moving the ransom money to them without identifying their bank accounts or location. Although the systems aren't perfectly anonymous, the money can move quickly enough through cooperating "laundering" sites to thwart law enforcement.

In a recent twist, the malware designers have found a way to use the cash transactions for a second purpose. The key that unlocks the victim's files, be it a symmetric key or a one-time private key, can be part of the transaction that pays the ransom. Bitcoin's block chain supports auxiliary transaction information, which is perfect for moving the key identifying information to the criminals and for letting them publish the symmetric or private key that unlocks the victim's files. The victim can attach the encrypted key blob or its identifier to the ransom payment, and the extortioner then puts the unlocked key into the transaction chain.

Other methods of delivering the decryption key are used. The ransomware can, for example, poll a command and control server. When payment is complete, that server will return the key to the victim's machine where, with any luck, the decryption will be completed quickly.

I haven't found any description of the methods used to verify payment and release the key. This must be a manual process, requiring the extortioner to communicate with a command and control server or to post the information in a public place. If law enforcement could infiltrate those processes, they might be able to release the data that unlocks the victim's machines.

Attacking New Platforms

Scott Adams' *Dilbert* cartoon on 12 May 2016 had the caption "My smartwatch was infected with ransomware" (http://dilbert.com/strip/2016-05-12). I laughed when I saw that, but experts warn that smartwatches are entirely hackable.[5] In fact, they're the harbingers of the world of smart and insecure wearables. The only saving grace is that these devices don't hold much data, and thus a factory reset should restore functionality.

While the attacks on digital accessories seem amusing now, the devices inexorably will acquire new features and importance in daily life. Our cellphones are becoming the linchpins of personal identity, reminders, and the way we contact other people. Unless we take care to provide offline storage for all this data, a ransomware attack could be devastating.

The major operating system providers take steps to insulate the various apps from one another's data, and this makes a complete takeover of a smartphone through a single compromised app unlikely. Nonetheless, all software has bugs, and a zero-day attack against a mobile OS kernel is sure to surface from time to time.

We can only hope that the designers of these gadgets realize their vulnerabilities and make sure that any essential data the gadgets hold is backed up with guards on the data's integrity and that it can be easily restored.

Offenses, Defenses

You're probably thinking that file backups are a simple way to defend yourself from ransomware. That's a good way to begin thinking about proactive measures, but the ransomware writers are way ahead of you. Unless you have a backup system that keeps copies of data offline and doesn't overwrite data for several weeks, you might still be vulnerable to ransomware. The malware designers methodically seek out backups, be they on the local machine's storage, on a shared file server, on a removable device, or in a cloud service.

When an afflicted machine has the ability to overwrite files on a shared server, all the files on the server are vulnerable to the crypto malware. Even one infected machine can destroy the files of a small business, for example.

Without a detailed understanding of how his files are backed up, a user might be at the mercy of ransomware. Some users have been dismayed to discover that their backups contained the encrypted files. This happens because when a file encryptor causes the file contents to change, backup system will notice the new version and will save it. To restore the unencrypted data, the user needs access to a backup that has been inaccessible to the malware and was written shortly before the malware began its work.

The unpleasant truth is that users need to understand their backup service in some detail before declaring themselves safe from crypto ransomware. They need to think about their backup service in terms of resilience from a concerted attack. When is a full backup done? Can it be deleted or overwritten without the user's explicit permission? How often are incremental backups done? Can they be deleted or overwritten?

Website administrators are usually at less of a risk, even though there's ransomware that targets them through http. The website content is usually stored on servers that aren't part of the website itself, and the content is uploaded to the servers. If the servers are hacked, the content can be easily restored from its normal repository.

As several people[7] have pointed out during the ongoing debates about encryption policy, almost all software has exploitable bugs, and ransomware is no exception. I would guess that given enough time, most skilled security firms could break any ransomware. The keys might be inadvertently exposed in the software, the public keys might have a lot of bits but be badly chosen, the encodings might leak data, the key generators could be faulty, or the command and control servers might be hackable.

Symantec researchers partially agree with that assessment when they state the following:

> *But even with improved encryption, some recent ransom schemes are still not always water tight. Poor operations and procedures dog the efforts of cybercriminals, leaving victims with room to maneuver. Even today, some still continue to make rookie mistakes such as leaving behind keys. This suggests that the current ransomware scene is highly fragmented with many new actors trying to establish themselves in a market already dominated by small groups of professional cybercriminals.*[5]

But most people don't have the luxury of doing without their data while the experts investigate. Paying the ransom might be the only practical solution. Further, there's reason to suspect that the skill level of ransomware developers is rising. Detailed examinations of two examples, zCrypt[8] and Maktub,[9] reveal sophisticated methods for evading detection while they encrypt files. Incidentally, zCrypt uses public key encryption on files and is therefore very slow. Strangely, it doesn't compensate by using Maktub's trick of compressing the files before encryption.

If ransomware continues its path toward a hardened, almost foolproof implementation, new methods of protection might be brought into play. The operational characteristics of encryption processes could be used against it. For example, the repetitive loop of the AES cipher could be detected by runtime execution monitors. The same is true of the large number of multiplications that RSA entails. Moreover, an encrypted file is radically different from a non-encrypted file. Most notably, the number of zeros and ones will be almost the same for an encrypted file, but ordinary files are unlikely to have such an even distribution. So theoretically you could devise an execution monitor that randomly sampled instruction traces in real time, and if encryption was happening in anything other than SSL or other "authorized" encryption program, the monitor would look at its open file descriptors to see if it was writing "gobbledygook" into an ordinary file.

The people behind ransomware seem to have a good grasp on a dangerous technology, and they've turned it into a profitable business. Although its delivery method is usually the antiquated trick of hiding malware in an email attachment, this remains effective and catches millions of people each year. Ransomware is becoming so notorious that one of the inventors of public key cryptography has said he feels like a parent whose child has become a terrorist.[10]

The cleverness of ransomware should be countered by a three-pronged approach. First, the delivery of malware through email attachments should be stomped out through better operating system protections on the major OSs. Second, backup services should specifically address ransomware through better retention times and protection from being written over or deleted by malware. And finally, the integrity of file system data should be the subject of more development. Malware shouldn't be able to write files.

Until the majority of computer systems (and that includes mobile devices) have these protections built-in, the ransomware industry seems likely to flourish.

References

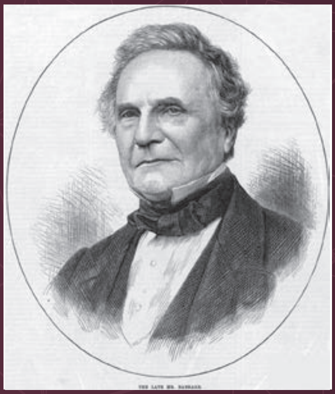
1. Sun Tzu, *The Art of War*; http://classics.mit.edu/Tzu/artwar.html.
2. Subcommittee on Networking and Information Technology Research and Development (NITRD), *Federal Cybersecurity Research and Development Strategic Plan: Ensuring Prosperity and National Security*, tech. report, US National Science and Technology Council (NSTC), 2015; www.whitehouse.gov/sites/whitehouse.gov/files/documents/2016_Federal_

Cybersecurity_Research_and_Development_Stratgeic_Plan.pdf.

3. K. Laffan, "A Brief History of Ransomware," blog entry, *Varonis*, 10 Nov. 2015; https://blog.varonis.com/a-brief-history-of-ransomware.
4. A. Young and M. Yung, "Cryptovirology: Extortion-Based Security Threats and Countermeasures," *Proc. IEEE Symp. Security and Privacy*, 1996, pp. 129–140.
5. K. Savage, P. Coogan, and H. Lau, *The Evolution of Ransomware*, version 1.0, white paper, Symantec, 6 Aug. 2015; www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/the-evolution-of-ransomware.pdf.
6. P. Ducklin, "TeslaCrypt Ransomware Gang Reveals Master Key to Decrypt Files," *Naked Security*, *Sophos.com*, 19 May 2016; https://nakedsecurity.sophos.com/2016/05/19/teslacrypt-ransomware-gang-shuts-up-shop-reveals-master-key/.
7. S. Bellovin et al., "Lawful Hacking: Using Existing Vulnerabilities for Wiretapping on the Internet," *Northwestern J. Technology and Intellectual Property*, vol. 12, no. 1, 2014; http://scholarlycommons.law.northwestern.edu/njtip/vol12/iss1/1.
8. MlwrHpstr, *ZCrypt Ransomware: Under the Hood*, blog, Malwarebytes Labs, 14 June 2016;https://blog.malwarebytes.com/threat-analysis/2016/06/zcrypt-ransomware.
9. Hasherezade, *Maktub Locker – Beautiful and Dangerous*, blog, Malwarebytes Labs, 24 Mar. 2016; https://blog.malwarebytes.com/threat-analysis/2016/03/maktub-locker-beautiful-and-dangerous.
10. J. Schwartz, "RSA Encryption Inventors Lament Its Use for Ransomware," blog, *Redmond Magazine*; 22 Apr. 2015; https://redmondmag.com/Blogs/The-Schwartz-Report/2015/04/Encryption-Ransomware.aspx.

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.

This article originally appeared in IEEE Internet Computing, *vol. 20, no. 5, 2016.*



Editor: **Michiel van Genuchten**
VitalHealth Software
genuchten@ieee.org

Editor: **Les Hatton**
Oakwood Computing Associates
lesh@oakcomp.co.uk

Creating the Virtual Universe

Simon Portegies Zwart and Jeroen Bédorf

In *The Hitchhiker's Guide to the Galaxy*, the guide's editor had a universe in his office. Now you can too. Hopefully, there's no Total Perspective Vortex in this one. —Michiel van Genuchten and Les Hatton

TO UNDERSTAND THE universe, modern science employs a three-pronged strategy: an empirical perspective, by gazing at the stars; a theoretical perspective, by developing conjectures; and simulation. In particular, computer simulations are essential for acquiring this understanding. That's because our view of the heavens is limited to a single perspective in a minuscule volume in space-time, and only a small portion of that universe fits in a lab. Simulation can be separated again into three areas: hardware, algorithms, and software engineering. Here, we focus on software engineering because it's often considered the least important.

The immensity of space and the fact that the smallest scales are intricately coupled to the largest scales make modeling the universe a challenge. We're just starting to understand how to combine microscopic scales with macroscopic scales in the computer. Our ability to reliably solve the physics for an extended range of scales is key to eventually understanding the universe.

Another challenge, though, might pose an even bigger problem: the intrinsic multiphysics aspect of the universe. By definition, all the physics we know, and all the physics we don't know, comes together here. This complexity is well posed in the *Hitchiker's Guide to the Galaxy*:

> *All you really need to know for the moment is that the universe is a lot more complicated than you might think, even if you start from a position of thinking it's pretty damn complicated in the first place.*[1]

On the basis of this quote, it seems that astronomers have accepted an impossible task. Nevertheless, to address these challenges and enable scientists to perform this task, we've developed simulation software we call the Astronomical Multipurpose Software Environment (AMUSE).[2]

Addressing the Challenges

Solving multiscale, multiphysics problems requires advanced computational techniques. The development of astronomical simulation software has always closely followed hardware advances.[3] But the traditional way of writing scientific software, as a single all-encompassing package, is ineffective and inefficient for the complexity at hand.

2469-7087/17/$33.00 © 2017 IEEE

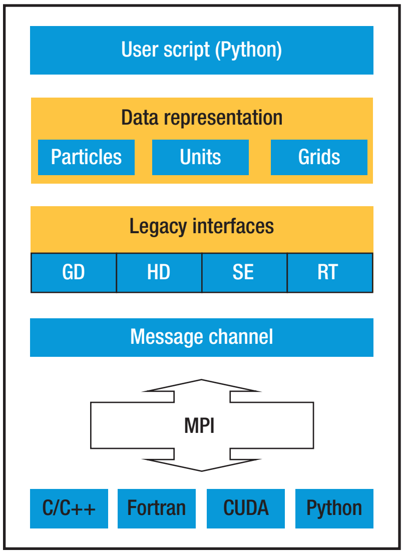


FIGURE 1. The structure of the Astronomical Multipurpose Software Environment (AMUSE). The bottom layer shows the (compiled) community codes. Current interfaces include those for gravitational dynamics (GD), hydrodynamics (HD), stellar evolution (SE), and radiative transfer (RT). CUDA stands for Compute Unified Device Architecture.

For example, scientific software development often starts as a PhD project. Between graduation and the eventual appointment to full professor, that software is either abandoned or still used. In the first case, the software is generally lost; in the second case, the package continues to grow. Either way, the software eventually becomes useless.

AMUSE deals with these issues in two ways. First, it keeps software packages simple; each package solves one type of physics on a limited scale. Second, it couples the packages at a higher level.

Two major developments helped give birth to AMUSE: the availability of optimized (monophysics) simulation software and the development of advanced scripting languages. Our technique for coupling the optimized, but computationally demanding, physics solvers with a rapid-prototyping management structure lets us address previously unaddressable astrophysics problems.

The bulk of the computation occurs in the packages, so we optimize them for high performance. The glue language we employ to blend the domain-specific packages is easy to use and understand and doesn't have to be highly optimized.

A Brief Look at AMUSE

AMUSE comprises an interface framework and 55 dedicated physics solvers for

- gravitational dynamics (19 solvers),
- stellar evolution (6 solvers),
- radiative transfer (7 solvers),
- hydrodynamics (5 solvers), and
- tasks such as analysis and generating initial conditions (18 solvers).

We call these solvers the *community codes*. They're written in various programming languages—in particular, Fortran 90 (18.5 percent of the LOC), C/C++ header files (16.2 percent), C++ (14.6 percent), C (12.9 percent), Fortran 77 (12.0 percent), CUDA (Compute Unified Device Architecture; 1.9 percent), and Java (1.0 percent). The interface framework is written in Python 2.7 (17.5 percent). On average, 27.6 percent of the 1.1 million code lines are comments with no particular trend across languages, and each file comprises 330 ± 100 lines.

Figure 1 shows AMUSE's structure, in which the physics interface and community codes (shown at the bottom) dominate. Figure 2 presents the evolution of the LOC.

Most of the community codes have been developed by different research groups, so the programming paradigms, styles, naming conventions, I/O, and so on are inconsistent. In addition, the vast majority of the codes are poorly documented. Some of the older codes originated in the 1960s and 1970s and continued to be developed until they were assimilated in AMUSE. The source codes represent diverse computational techniques, methods, algorithms, and physical understanding.

The community codes remain untouched; they communicate with the framework via MPI channels. This guarantees they give identical results when run separately and lets us assign a separate process to each code to prevent global naming conflicts. This strategy preserves the codes' low-level optimization and parallelization. The interface compiler automatically generates the interface code.

Each code uses its own set of units. To accommodate a more astrophysical feel, we introduced automatic unit conversion (the second layer in Figure 1). This conversion ensures that units in the script and the respective codes are consistent. This abstraction is incorporated in the interface layer, so even novice astronomy students can easily write simple scripts.

AMUSE's biggest advantage is its ability to combine physics-specific solvers hierarchically to create a complex environment for addressing multiscale, multiphysics problems. This capacity lets users construct complex, efficient simulation code and thus opens up a novel way to perform astronomical simulations.

Optimizing a Community Code

Many community codes are optimized for general or specific architectures. They remain optimized as intended even when used with less optimized codes.

One such optimized code is Bonsai, which is used for solving Newton's motion equations.[4] We named the code Bonsai because it has a small footprint in terms of LOC and it uses the Barnes-Hut tree algorithm to calculate gravitational forces.[5]

The community codes and the AMUSE framework exchange the relevant information at checkpoints in time without mutual awareness. This lets us combine Bonsai with solvers for hydrodynamics, stellar evolution, and radiative transfer without affecting each solver's performance, to the limit at which Amdahl's law[6] prevents the optimization of combined solvers beyond the slowest solver.

Bonsai is only one of over a dozen gravity solvers in AMUSE, each of which has slightly different characteristics. Figure 3 presents a rendition of our 200-billion-particle simulation of the Milky Way using Bonsai.

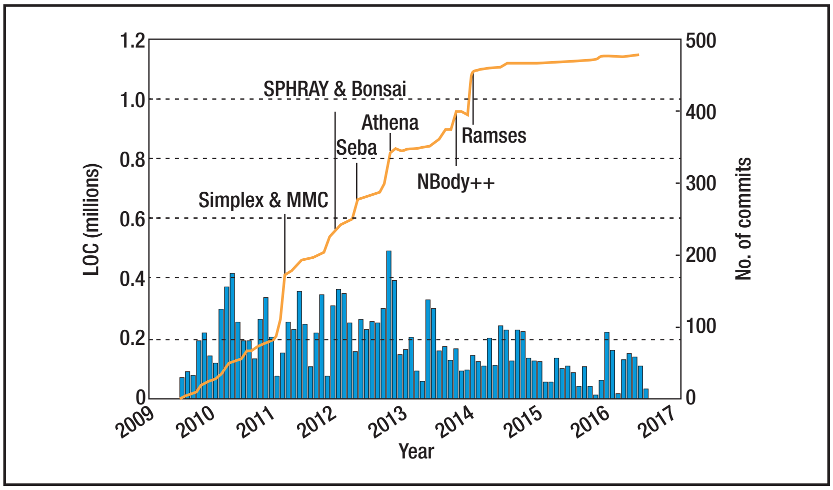


FIGURE 2. The evolution of the LOC (the solid curve) and commits (the histogram) for AMUSE. The text labels for the curve identify packages whose introduction caused jumps in the LOC. Since the end of 2014, we haven't added any major community codes; we've been focusing on improving usability, validation, and maintenance. SPHRAY stands for smoothed particle hydrodynamics ray tracer; MMC stands for Mocca Monte Carlo.

Bonsai's Design

Bonsai is a gravitational tree code, which means it uses a hierarchical tree structure to compute the gravitational force between sets of particles. This tree structure reduces the complexity of computing the force on a set of particles from $O(N^2)$ (direct N-body) to $O(N \log N)$. However, this computation is an approximate method that surrenders accuracy for speed. To tune this accuracy loss, we use the "opening angle" (θ); a wider angle results in faster computing at lower accuracy. (Strictly speaking, $\theta \rightarrow 0$ reduces the tree code to a rather inefficient direct N-body solver.)

Since the hierarchical-tree method's introduction in 1986, it has found many incarnations and applications in chemistry, molecular dynamics, oceanography, and so on. The algorithm's flexibility allows for enormous diversity and poses interesting optimization challenges.

Single-GPU Optimization

We started developing Bonsai after several years of experience developing direct N-body codes on single GPUs,[7] and we had extensive experience in parallel and distributed algorithms.[8] With the individual time-step integrator we were using, a CPU–GPU combined solver wouldn't scale satisfactorily. The frequent communication between the CPU and GPU required too much overhead, mainly because the local data structure was required throughout the tree code.

We took the radical design decision to port each algorithm to the GPU, including tree construction and traversing, the force-moment and multiple-moment computations, and N-body integration. This reduced communication between the CPU and GPU to a linear operation in the amount of data, which can easily be hidden in the computation.

For the tree-traversing and gravity calculations, which use the most computer cycles, we wrote a separate implementation for each generation of GPUs. This let us benefit from the latest hardware features without affecting the code's design.

Certain operations, such as integrating the motion equations, ported naturally to the GPU, whereas other operations, such as tree construction, had to be redesigned. We reduced tree construction to a bandwidth-limited operation by adopting prefix sums to detect the tree–node boundaries when particles are sorted along a Peano-Hilbert space-filling curve.[9]

With this optimization, the GPU did all the work, leaving the CPU available to handle multinode communication and runtime data analysis.

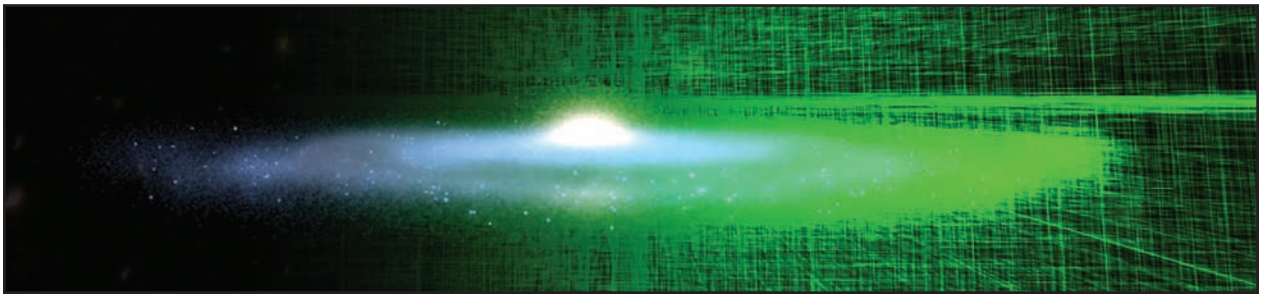


FIGURE 3. A rendition of the 200-billion-particle simulation of the Milky Way. This simulation employed Bonsai, a solver for Newton's motion equations.

Multinode Optimization

Then, we parallelized Bonsai over multiple GPUs. Because we built the parallel version on top of the sequential version, all previous optimizations naturally propagated to the multi-GPU version.

By using the CPU cores to streamline interaction with the GPU and network activities, we hide all the communication in the GPUs' computational workload. We use the leftover CPU performance for data processing, which is an irregular task with varying workloads and therefore not very suitable for the GPU.

We've run Bonsai simulations on small laptop GPUs, GPU-equipped workstations, local computer clusters, the 5,200 nodes of the Swiss National Computing Center's Piz Daint supercomputer, and the 18,600 nodes of Oak Ridge National Laboratory's Titan supercomputer, with over 85 percent efficiency.

Code Development and Validation

Forty-nine people have contributed to AMUSE, and at least as many people have contributed to the community codes. More than 120 example scripts demonstrate specific operations, and we carry out nightly unit tests of the fundamental interface operations (853 tests), basic functionality (247 tests), and individual community codes (1,154 tests). The repository is publicly available (on Github), but despite all the tests, maintaining the framework remains challenging, particularly owing to the diverse languages and styles.

To maintain performance, AMUSE is profiled with the standard Python profiler, but several community codes have their own profiling logistics. For example, Bonsai has integrated profiling that logs the performance and communication characteristics. After Bonsai performs a calculation, this logging data is analyzed—for example, to study the differences between various CUDA versions.

Validating a monophysics solver can be difficult, if not impossible. We can repeat the same simulations with a higher resolution, hoping that the solution converges, or we can compare the numerical solution with analytic results. AMUSE can perform those tests and more. It lets us seamlessly replace one code with another code that solves the same physics. In that way, we can directly compare one code's results with those of a similar code under identical conditions. This unique capability turns AMUSE into an excellent environment to verify individual codes.

Validating multiphysics solutions is and will remain difficult. For problems without an analytic solution or any other codes to compare with, validation and verification must be careful and thorough. So, each new multiphysics simulation requires a new set of validations. The possibility to test individual ingredients separately and replace specific community codes enables thorough testing, but there's no golden rule of how to do this. Nevertheless, AMUSE makes such testing much easier than hitherto.

Large-scale simulation software remains extremely hard to maintain and delicate in its use. We think that it will eventually adapt a distributed architecture whose components are dedicated, highly optimized, and small (in terms of LOC and the number of tasks). Interaction between these components can then be realized with a rapid-prototyping language such as Python.

We developed this concept in AMUSE. We're porting the AMUSE approach to oceanography and long-term weather prediction research. Although weather sounds like a different problem than black-hole dynamics in galactic nuclei, the funda-

This article originally appeared in IEEE Software, *vol. 33, no. 5, 2016.*

mental multiscale and multiphysics aspects remain similar.

AMUSE's diversity makes its maintenance challenging, and validation and verification remain concerns that must be reevaluated with each new combination of solvers. This is somewhat relaxed by AMUSE's unit conversion and transparency, but validation of a complex multidomain solver will require continuous attention.

The combination of highly optimized solvers for specific tasks and a general framework has worked excellently for multiscale and multidomain simulations. Regardless of the limitations and drawbacks, we think that the AMUSE approach has a bright future. For a comparison of our research to other research reported in *IEEE Software*'s Impact department, see the sidebar.

OUR RESEARCH AND PREVIOUS IMPACT ARTICLES

In the main article, we describe the Astronomical Multipurpose Software Environment (AMUSE), large-scale simulation software written by scientists. As such, AMUSE is most similar to the software behind the Higgs boson discovery, which was developed by a large group of physicists.[1]

AMUSE also has an interesting similarity with Michiel van Malkenhorst and Lex Mollinger's software for dynamic oil exploration:[2] verifying a simulation's correctness is difficult. As van Malkenhorst and Mollinger stated, "But a more dangerous type of defect exists: faulty physics .... But what is reality for something you can't see otherwise?" Oil explorers can eventually find out whether their simulation is correct by drilling. With astrophysics, the scientific community decides which results are valid.

In 2012, Michiel van Genuchten and Les Hatton calculated an average compound annual growth rate (CAGR) of 1.16 for the software described in six Impact articles.[3] From 2010 to 2016, AMUSE has grown from 150 KLOC to 1.1 million LOC, representing a CAGR of 1.4. This faster growth rate is due to the inclusion of the relatively independent modules we've developed over the past decades.

References

1. D. Rousseau, "The Software behind the Higgs Boson Discovery," *IEEE Software*, vol. 29, no. 5, 2012, pp. 11–15.
2. M. van Malkenhorst and L. Mollinger, "Going Underground," *IEEE Software*, vol. 29, no. 3, 2012, pp. 17–20.
3. M. van Genuchten and L. Hatton, "Quantifying Software's Impact," *Computer*, vol. 46, no. 10, 2013, pp. 66–72.

Acknowledgments

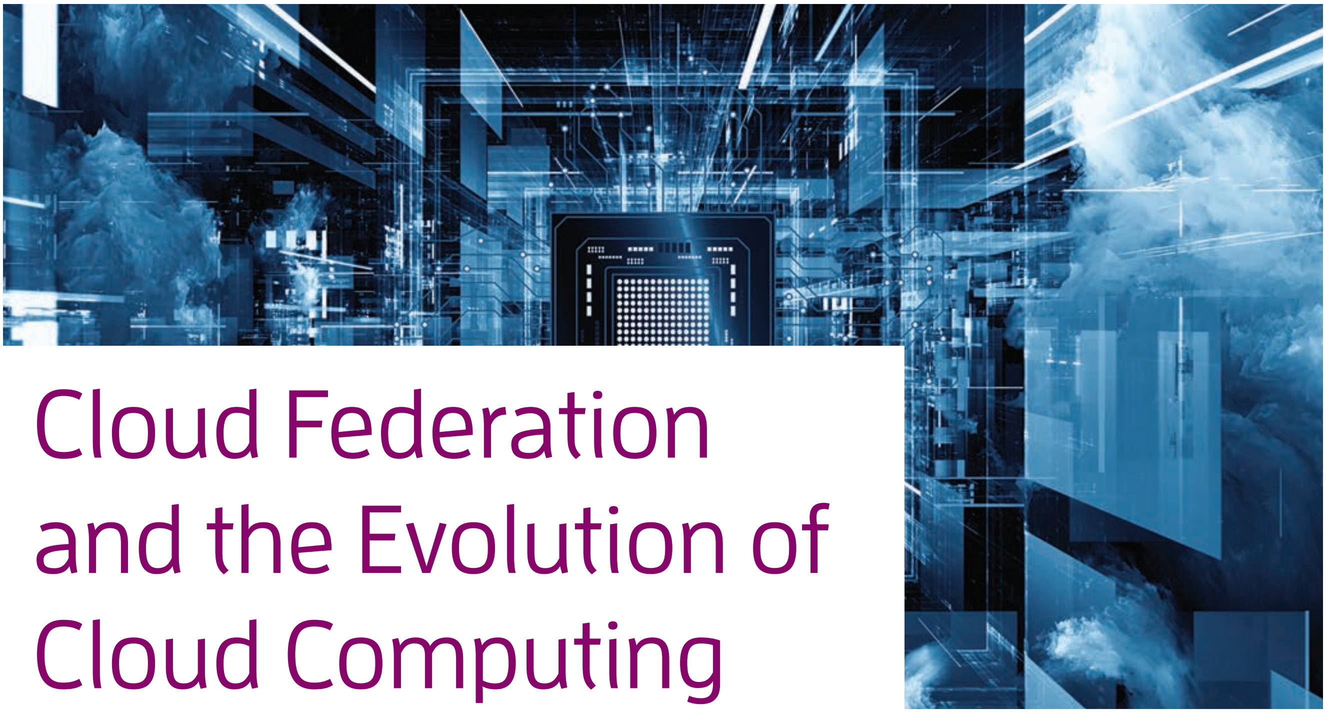
Creating the Astronomical Multipurpose Software Environment would have been impossible without the invaluable help of Arjen van Elteren and Inti Pelupessy. Jack Wells enabled us to test our Bonsai code on Oak Ridge National Laboratory's Titan supercomputer.

References

1. D. Adams, *The Hitchhiker's Guide to the Galaxy*, Pan Books, 1979.
2. S.F. Portegies Zwart et al., "Multiphysics Simulations Using a Hierarchical Interchangeable Software Interface," *Computer Physics Comm.*, vol. 184, no. 3, 2013, pp. 456–468.
3. S.F. Portegies Zwart and J. Bédorf, "Using GPUs to Enable Simulation with Computational Gravitational Dynamics in Astrophysics," *Computer*, vol. 48, no. 11, 2015, pp. 50–58.
4. J. Bédorf et al., "24.77 Pflops on a Gravitational Tree-Code to Simulate the Milky Way Galaxy with 18600 GPUs," *Proc. 2014 Int'l Conf. High Performance Computing, Networking, Storage and Analysis* (SC 14), 2014, pp. 54–65.
5. J. Barnes and P. Hut, "A Hierarchical O(*N* log *N*) Force-Calculation Algorithm," *Nature*, vol. 324, 1986, pp. 446–449.
6. G.M. Amdahl, "Validity of the Single Processor Approach to Achieving Large Scale Computing Capabilities," *Proc. 1967 Spring Joint Computer Conf.* (AFIPS 67 (Spring)), 1967, pp. 483–485.
7. S.F. Portegies Zwart, R.G. Belleman, and P.M. Geldof, "High-Performance Direct Gravitational *N*-Body Simulations on Graphics Processing Units," *New Astronomy*, vol. 12, no. 8, 2007, pp. 641–650.
8. D. Groen et al., "Distributed *N*-Body Simulation on the Grid Using Dedicated Hardware," *New Astronomy*, vol. 13, no. 5, 2008, pp. 348–358.
9. G. Peano, "Sur une courbe, qui remplit toute une aire plane" [On a Curve That Fills a Flat Area], *Mathematische Annalen*, vol. 36, no. 1, 1890, pp. 157–160.

SIMON PORTEGIES ZWART is a professor of computational astrophysics at the Leiden Observatory. Contact him at spz@strw.leidenuniv.nl.

JEROEN BÉDORF is a postdoc at the Leiden Observatory. Contact him at jeroen@bedorf.net.



Cloud Federation and the Evolution of Cloud Computing

Dimitrios G. Kogias, Michael G. Xevgenis, and Charalampos Z. Patrikakis,
Piraeus University of Applied Sciences

To satisfy the demand for collective and collaborative cloud use, academia and industry want to interconnect heterogeneous clouds to form a federated system. This approach is promising but also faces significant challenges.

Cloud computing allows users to access computing services and resources on demand without having to buy their own infrastructures, and to pay only for what they use.[1] Many cloud companies—such as Amazon and Google—have developed their own platforms featuring proprietary interfaces, which isn't a problem as long as a single provider can fully satisfy its customers. However, the lack of standardization for interconnecting platforms makes it difficult for customers who need the combined services or resources of multiple providers. This often results in users being locked into specific providers and platforms.[2,3]

This issue has led to the idea of interconnected clouds, also known as *interclouds*.[2–5] Interclouds address single-provider approaches' limitations such as the lack of interoperability between platforms, limited resources being exhausted during times of peak customer demand, service interruptions, and quality-of-service (QoS) degradation.

INTERCLOUD

An intercloud is a cloud of clouds.[3] In essence, it's a large cloud comprising many smaller clouds, each having its own characteristics and serving different needs. An intercloud implementation could be any one or combination of

- *hybrid clouds*, in which private clouds access the resources of public clouds without the latter being aware of their participation;
- *multiclouds*, which utilize libraries from applications that enable the use of resources from multiple clouds, without any of them being aware of their participation;
- *sky computing*, an emerging model in which resources from multiple cloud service providers (CSPs) create a large, distributed, virtual infrastructure

2469-7087/17/$33.00 © 2017 IEEE

able to support the establishment of trust between different clouds that might not be configured to trust or even recognize one another;[6]
- *multiclouds tournament*, an architecture comprising multiple clouds that utilizes a tournament model to balance resource offerings with users' consumption, thereby providing higer-quality services;[7] and
- *cloud federations*, an interconnected set of heterogeneous public and/or private clouds from voluntarily participating users and providers.[2,3]

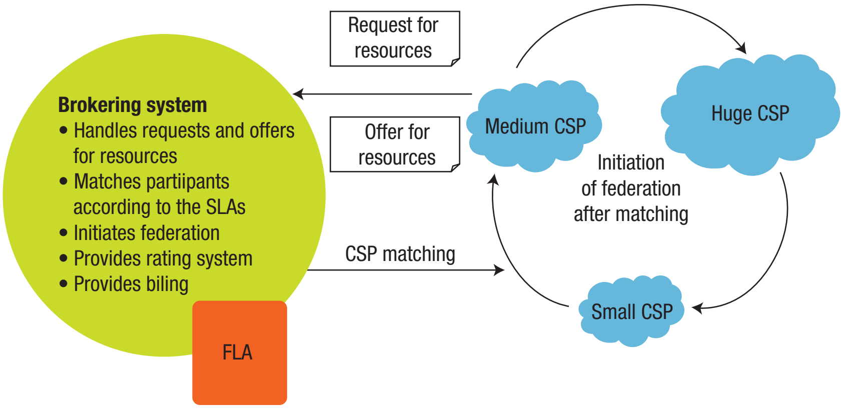


Figure 1. Cloud federation architecture. Users send requests for resources and cloud service providers (CSPs) send their responses to the broker (left), which matches users with providers based on billing, ratings, and service-level agreements (SLAs). This results in a federation (right), governed by a federal-level agreement (FLA).

CLOUD FEDERATION

Intercloud researchers have shown the most interest in cloud federations because it enables power-efficient, cost-effective, dynamic sharing of idle cloud resources and services. Federation members can sign service-level agreements (SLAs) to ensure QoS and availability.

The federation should

- have a defined marketing system that describes the cost of utilizing resources and services and that helps to valorize use,
- feature efficient geographic dispersion by allocating resources close to users to eliminate network problems that could interrupt service access, and
- follow rules in a federal-level agreement (FLA) describing the cooperation and relationship among participating clouds.

We disagree with the research literature's frequent interchangeable use of the terms "cloud federation" and "intercloud." In federations, cloud organizations participate voluntarily after signing an FLA. In an intercloud organization, no private or public cloud is necessarily aware of its participation. Also, interclouds are based on open standards that provide interfaces for interoperability. Cloud federations use a broker to translate and connect CSPs' own interfaces.

CLOUD FEDERATION ARCHITECTURE

For federations or interclouds to work properly, heterogeneous clouds must be able to interoperate. However, this can be difficult to achieve. For example, participating clouds might use different techniques to describe the services they offer. Users, however, need a mechanism to provide common access to available services. Thus, the cloud federation's architecture must employ interface standards, a service broker that translates between interfaces and provides updates on offered services and users' status changes, or a combination of the two.[3,8]

Cloud federations most often use brokerages. The common object request broker architecture (CORBA) and object request broker (ORB) middleware were initially the most popular approaches.[9] However, the advent of XML-based technologies such as SOAP has provided the ability to use the same language in the descriptions of all services, thereby avoiding the need for translation.

Figure 1 shows a cloud federation architecture with the broker playing a central role and the CSPs at the edges communicating mainly through the broker. The brokering system is in the cloud and matches the available federation resources with user demand, taking into consideration participants' SLAs. To achieve this, the broker must understand the various ways that each cloud describes its available resources and services[2,3] and then combine the gathered information seamlessly for the user. In some cases, the broker could provide users with resource and service pricing information, as well as bill them.

For the federation to function properly, all interested parties must sign an FLA that specifies interconnection rules and describes each participant's responsibilities and permissible behaviors, along with the financial, administrative, or other penalties for violating its terms. The parties can leave the federation when they want, as long as they follow FLA procedures.

ADVANTAGES AND LIMITATIONS OF CLOUD FEDERATION

Cloud federations have pros and cons.

Advantages

Federation performance is guaranteed by the dynamic resource allocation—or *elasticity*—that lets clouds ask for other participants' idle resources or services when their own are exhausted. This achieves both uninterrupted service delivery and resource scalability, the latter being the result of the seamless, transparent operation between clouds for the delivery of an agreed-upon QoS level.

Federations also enable the geographic dispersion of resources, efficiently locating some near users[10] but also allowing participants to access more distant resources in case of local outages. This enables efficient commercialization of the offered resources and lower prices than single-cloud services can charge.[11]

And because the FLA clearly describes what each participant is offering, as well as the federation's rules, it ensures the commitment of the involved parties to the operation's performance.

Limitations

Although federation mechanisms can provide the agreed-upon performance, constant monitoring and increased security mechanisms are required to guard against accidents and malicious users.

Selecting which services a federation will offer is not trivial because they will have to come from multiple providers that have different cloud characteristics and that offer varying QoS levels. Thus, federation participations should deploy a service-selection mechanism, preferably automated, that uses a predefined set of criteria regarding the QoS that providers offer. Or they could dynamically negotiate SLAs to address user needs.

Federation members could also address the lack of a common repository for available services via peer-to-peer approaches using a distributed hash table overlay network for service discovery.[12] They could also utilize an intercloud root,[13] which produces an abstract view of a global catalog of federation services and resources offered in the connected clouds.

Cloud federations enable power-efficient, cost-effective, dynamic sharing of cloud providers' idle resources and services. This approach could promote more collaborative use of the cloud, but it also faces significant challenges.

The mobility of virtual machines (VMs), which are common in cloud services, is important for providing uninterrupted performance and expected QoS levels. Hosts must meet requirements for factors such as memory use, state, status of running processes and applications, and LAN connectivity to be able to migrate a live VM from one physical node to another without disrupting network traffic. This is particularly critical in real-time services. In cloud environments, this migration could be challenging for VMs belonging to different clouds that have never shared resources and thus have no knowledge about each other's networking configurations. Thus, it's important to re-create the originating cloud's networking and communication environment in the destination cloud quickly enough to avoid excessive delays.

Federation participants must address data portability, focusing particularly on issues such as security and privacy, because services belonging to one CSP must frequently access data stored in another cloud.

LOOKING AHEAD

Early attempts at cloud federations haven't had all the characteristics that a true federation should possess. Instead, there have been multiclouds or hybrid clouds enhanced with some federation characteristics. However, these aren't as efficient as fully federated approaches.

True federations require brokering systems that can quickly communicate with cloud interfaces and find the right combination of resources and QoS to meet users' needs in the heterogeneous environment. In the process, the brokerages must keep in mind users' performance and cost requirements.

Content delivery networks (CDNs)—which have successfully provided high-quality data access for many users over the Internet—could serve as the framework for cloud-broker communication. But regardless of which approach is adopted, the CSPs' role is important, particularly for providing APIs that enable communication with brokers. Standards organizations such as IEEE could also play a major role in cloud-federation evolution by developing a reliable brokering system that is compatible with most cloud frameworks.

Federation participants must take special care in composing the terms of an FLA, which is the mechanism that ensures the system's integrity. A key concern is translating abstractly expressed requirements into concrete technical terms and functionalities.

Other issues include the establishment of trust among participants and the security of resource access and use, which is extremely important in a dynamic environment such as a cloud federation.

IEEE's effort[14] to introduce a standard for a brokering-system is an important step toward the realization of true cloud federations. Researchers should also examine the characteristics proposed in different cloud technologies and architectures—such as fog computing's local hardware awareness[15]—that provide the technical capabilities that VMs could use to learn about cloud environments.

CALL FOR COLUMN CONTRIBUTIONS

For this column, we welcome short articles (1,500 to 2,000 words) discussing your ideas for advancing cloud computing or sharing your experiences in harnessing the cloud. We also solicit articles on recent developments, future trends, case studies, and topics such as cloud governance, cloud management and monitoring, risk management, disaster recovery, open source cloud computing, pricing, cloud economics, service-level agreements, standards, compliance, and legal issues. Please send proposals or submissions to San Murugesan at cloudcover@computer.org. For a list of previous Cloud Cover columns, visit http://tinyurl.com/computer-cloudcover.

REFERENCES

1. *Encyclopedia of Cloud Computing*, S. Murugesan and I. Bojanova, eds., Wiley-IEEE Press, 2016.
2. M.R.M. Assis and L.F. Bittencourt, "A Survey on Cloud Federation Architectures: Identifying Functional and Non-functional Properties," *J. Network and Computer Applications*, vol. 72, September 2016, pp. 51–71.
3. A.N. Toosi, R.N. Calheiros, and R. Buyya, "Interconnected Cloud Computing Environments: Challenges, Taxonomy, and Survey," *ACM Computing Surveys*, vol. 47, no. 1, 2014, pp. 7:1–7:47.
4. D. Bernstein and Y. Demchenko, "The IEEE Intercloud Testbed—Creating the Global Cloud of Clouds," *Proc. IEEE 5th Int'l Conf. Cloud Computing Technology and Science* (CloudCom 13), 2013, pp. 45–50.
5. B. Di Martino et al., "Towards an Ontology-Based Intercloud Resource Catalogue—The IEEE P2302 Intercloud Approach for a Semantic Resource Exchange," *Proc. IEEE Int'l Conf. Cloud Eng.* (IC2E 15), 2015, pp. 458–464.
6. K. Keahey et al., "Sky Computing," *IEEE Internet Computing*, vol. 13, no. 5, 2009, pp. 43–51.
7. M.R.M. Assis, L.F. Bittencourt, "Multiclouds Tournament Blueprint," *Proc. IEEE/ACM 8th Int'l Conf. Utility and Cloud Computing* (UCC 15), 2015, pp. 404–405.
8. G. Zangara et al., "A Cloud Federation Architecture," *Proc. 10th Int'l Conf. P2P, Parallel, Grid, Cloud, and Internet Computing* (3PGCIC 15), 2015, pp. 498–503.
9. M. Henning, "The Rise and Fall of CORBA," *ACM Queue*, vol. 4, no. 5, 2006, pp. 28–34.
10. L. Hongxing et al., "Virtual Machine Trading in a Federation of Clouds: Individual Profit and Social Welfare Maximization," *IEEE/ACM Trans. Networking*, vol. 24, no. 3, 2016, pp. 1827–1840.
11. J. Weinman, "Intercloudonomics: Quantifying the Value of the Intercloud," *IEEE Cloud Computing*, vol. 2, no. 5, 2015, pp. 40–47.
12. R. Ranjan and L. Zhao, "Peer-to-Peer Service Provisioning in Cloud Computing Environments," *J. Supercomputing*, vol. 65, no. 1, 2013, pp. 154–184.
13. D. Bernstein and D. Vij, "Intercloud Directory and Exchange Protocol Detail Using XMPP and RDF," *Proc. 6th World Congress on Services* (Services 10), 2010, pp. 431–438.
14. *IEEE P2302/D0.2 Draft Standard for Intercloud Interoperability and Federation (SIIF)*, IEEE, 2012; www.intercloudtestbed.org/uploads/2/1/3/9/21396364/intercloud_p2302_draft_0.2.pdf.
15. M. Zhanikeev, "A Cloud Visitation Platform to Facilitate Cloud Federation and Fog Computing," *Computer*, vol. 48, no. 5, 2015, pp. 80–83.

DIMITRIOS G. KOGIAS is an adjunct lecturer and a senior researcher in the Piraeus University of Applied Sciences' Department of Electronics Engineering. Contact him at dimikog@teipir.gr.

MICHAEL G. XEVGENIS is a junior researcher in Piraeus University of Applied Sciences' Department of Electronics Engineering. He is also a postgraduate student at Kingston University. Contact him at mxevgenis@teipir.gr.

CHARALAMPOS Z. PATRIKAKIS is an associate professor in the Piraeus University of Applied Sciences' Department of Electronics Engineering. Contact him at bpatr@teipir.gr.

This article originally appeared in Computer, *vol. 49, no. 11, 2016.*

Congestion on the Last Mile

Shane Greenstein
Harvard Business School

It has long been recognized that networked services contain weak-link vulnerabilities. That is, the performance of any frontier device depends on the performance of every contributing component and service. This column focuses on one such phenomenon, which goes by the label "congestion." No, this is not a new type of allergy, but, as with a bacteria, many users want to avoid it, especially advanced users of frontier network services.

Congestion arises when network capacity does not provide adequate service during heavy use. Congestion slows down data delivery and erodes application performance, especially for time-sensitive apps such as movies, online videos, and interactive gaming.

Concerns about congestion are pervasive. Embarrassing reports about broadband networks with slow speeds highlight the role of congestion. Regulatory disputes about data caps and pricing tiers question whether these programs limit the use of data in a useful way. Investment analysts focus on the frequency of congestion as a measure of a broadband network's quality.

What economic factors produce congestion? Let's examine the root economic causes.

The Basics

Congestion arises when demand for data exceeds supply in a very specific sense.

Start with demand. To make this digestible, let's confine our attention to US households in an urban or suburban area, which produce the majority of data traffic.

No simple generalization can characterize all users and uses. The typical household today uses data for a wide variety of purposes—email, video, passive browsing, music videos, streaming of movies, and e-commerce. Networks also interact with a wide variety of end devices—PCs, tablets, smartphones on local Wi-Fi, streaming to television, home video alarm systems, remote temperature control systems, and plenty more.

It is complicated, but two facts should be foremost in this discussion. First, a high fraction of traffic is video—anywhere from 60 to 80 percent, depending on the estimate. Second, demand peaks at night. Most users want to do more things after dinner, far more than any other time during the day.

Every network operator knows that demand for data will peak (predictably) between approximately 7 p.m. and 11 p.m. Yes, it is predictable. Every day of the week looks like every other, albeit with steady growth over time and with some occasional fluctuations for holidays and weather. The weekends don't look any different, by the way, except that the daytime has a bit more demand than during the week.

The bottom line: evenings require far greater capacity than other times of the day. If capacity is not adequate, it can manifest as a bottleneck at many different points in a network—in its backbone, in its interconnection points, or in its last mile nodes.

This is where engineering and economics can become tricky to explain (and to manage). Consider this metaphor (with apologies to network engineers): metaphorically speaking, network congestion can resemble a bathtub backed up with water. The water might fail to drain because something is interfering with the mouth of the drain or there is a clog far down the pipes. So, too, congestion in a data network can arise from inadequate capacity close to the household or inadequate capacity somewhere in the infrastructure supporting delivery of data.

Numerous features inside a network can be responsible for congestion, and that shapes which set of households experience congestion most severely. Accordingly, numerous different investments can alleviate the congestion in specific places. A network could require a "splitting of nodes" or a "larger pipe" to support a content delivery network (CDN) or could require "more ports at the point of interconnection" between a particular backbone provider and the network.

As it turns out, despite that complexity, we live in an era in which bottlenecks arise most often in the last mile, which ISPs build and operate. That simplifies the economics: once an ISP builds and optimizes a network to meet maximum local demand at peak hours, then that same capacity will be able to meet lower

2469-7087/17/$33.00 © 2017 IEEE

This article originally appeared in IEEE Micro, *vol. 36, no. 6, 2016*

demand the rest of the day. Similarly, high capacity can also address lower levels of peak demand on any other day.

Think of the economics this way. An awesome network, with extraordinary capacity optimized to its users, will alleviate congestion at most households on virtually every day of the week, except the most extraordinary. Accordingly, as the network becomes less than awesome with less capacity, it will generate a number of (predictable) days of peak demand with severe congestion throughout the entire peak time period at more households. The logic carries through: the less awesome the network, the greater the number of households that experience those moments of severe congestion, and the greater the frequency.

That provides a way to translate many network engineering benchmarks—such as the percentage of packet loss. More packet loss correlates with more congestion, and that corresponds with a larger number of moments when some household experiences poor service.

Tradeoffs and Externalities

Not all market participants react to congestion in the same way. Let's first focus on the gazillion Web firms that supply the content. They watch this situation with a wary eye, and it's no wonder. Many third-party services, such as those streaming video, deliver a higher-quality experience to users whose network suffers less congestion.

Many content providers invest to alleviate congestion. Some invest in compression software and superior webpage design, which loads in ways that speed up the user experience. Some buy CDN services to speed delivery of their data. Some of the largest content firms, such as YouTube, Google, Netflix, and Facebook, build their own CDN services to improve delivery.

Next, focus on ISPs. They react with various investment and pricing strategies. At one extreme, some ISPs have chosen to save money by investing conservatively, and they suffer the complaints of users. At the other extreme, some ISPs build a premium network, then charge premium prices for the best services.

There are two good reasons for that variety. First, ISPs differ in their rates of capital investment. Partly this is due to investment costs, which vary greatly with density, topography, and local government relations. Rates of investment tend to be inherited from long histories, sometimes as a product of decisions made many years ago, which accumulated over time. These commitments can change, but generally don't, because investors watch capital commitments and react strongly to any departure from history.

The second reason is more subtle. ISPs take different approaches to raising revenue per household, and this results in (effectively) different relationships with banks and stockholders, and, de facto, different budgets for investment. Where does the difference in revenue come from? For one, competitive conditions and market power differ across neighborhoods. In addition, ISPs use different pricing strategies, taking substantially different approaches to discounts, tiered pricing structures, data cap policies, bundled contract offerings, and nuisance fees.

The use of tiers tends to grab attention in public discussion. ISPs segment their users. Higher tiers bring more bandwidth to a household. All else equal, households with higher tiers experience less congestion at peak moments.

Investors like tiers because they don't obligate ISPs to offer unlimited service and, in the long run, they raise revenue without additional costs. Users have a more mixed reaction. Light users like the lower prices of lower tiers, and appreciate saving money for doing little other than email and static browsing. In contrast, heavy users perceive that they pay extra to receive the bandwidth that the ISP used to supply as a default.

ISPs cannot win for losing. The archetypical conservative ISP invests adequately to relieve congestion some of the time, but not all of the time. Its management then must face the occasional phone calls from its users, which they stymie with phone trees that make service calls last 45 minutes. Even if users like the low prices, they find the service and reliability quite irritating.

The archetypical aggressive ISP, in contrast, achieves a high-quality network, which relieves severe congestion much of the time. Yet, such firms (typically) find clever ways to pile on fees, and know how to stymie user complaints with a different type of phone tree that makes calls last 45 minutes. Even when users like the quality, the aggressive pricing practices tend to be quite irritating.

One last note: it is a complicated situation where ISPs interconnect with content providers. Multiple parties must invest, and the situations involve many supplier interests and strategic contingencies.

Some observers have alleged that the biggest ISPs have created congestion issues at interconnection points for purposes of gaining negotiating leverage. These are serious charges, and a certain amount of skepticism is warranted for any broad charge that lacks specifics. Somebody ought to do a sober and detailed investigation to confront those theories with evidence. (I am just saying.)

What does basic economics tell us about congestion? Congestion is inevitable in a network with interlocking interests. When one part of the network has congestion, the rest of it catches a cold.

More to the point, growth in demand for data should continue to stress network capacity into the foreseeable future. Since not all ISPs will invest aggressively in the presence of congestion, some amount of congestion is inevitable. So, too, is a certain amount of irritation. MICRO

Shane Greenstein is a professor at the Harvard Business School. Contact him at sgreenstein@hbs.edu.

IEEE Computer Society Is Where You Choose the Resources that Fit Your Career

Find the membership that fits you best. IEEE Computer Society lets you choose your membership — and the benefits it provides — to meet your specific career needs. With four professional membership categories and one student package, you can select the precise industry resources, offered exclusively through the Computer Society, that will help you achieve your goals.

Select your membership	Preferred Plus		Training & Development		Research		Basic		Student
	$60 *IEEE Member*	**$126** *Affiliate Member*	**$55** *IEEE Member*	**$115** *Affiliate Member*	**$55** *IEEE Member*	**$115** *Affiliate Member*	**$40** *IEEE Member*	**$99** *Affiliate Member*	**$8** *Does not include IEEE membership*
Computer magazine (12 digital issues)*	✓		✓		✓		✓		✓
ComputingEdge magazine (12 issues)	✓		✓		✓		✓		✓
Members-only discounts on conferences and events	✓		✓		✓		✓		✓
Members-only webinars	✓		✓		✓		✓		✓
Unlimited access to *Computing Now*, computer.org, and the new mobile-ready myCS	✓		✓		✓		✓		✓
Local chapter membership	✓		✓		✓		✓		✓
Safari Books Online (600 titles and 50 training videos)	✓		✓						✓
Skillsoft online solutions (courses, certifications, practice exams, videos, mentoring)	✓		✓						✓
Two complimentary Computer Society magazine subscriptions	✓				✓				
myComputer mobile app	*30 tokens*				*30 tokens*				*30 tokens*
Computer Society Digital Library	*12 FREE downloads*		*Member pricing*		*12 FREE downloads*		*Member pricing*		*Included*
Training webinars	*3 FREE webinars*		*3 FREE webinars*		*Member pricing*		*Member pricing*		*Member pricing*
Priority registration to Computer Society events	✓								
Right to vote and hold office	✓		✓		✓		✓		
One-time 20% Computer Society online store discount	✓								

* *Print publications are available for an additional fee. See catalog for details.*

www.computer.org/membership

IEEE computer society

Careers in Cybersecurity Technology

For this *ComputingEdge* issue, we asked Dan Haagman—cybersecurity entrepreneur and cofounder of NotSoSecure Global Services, a leading UK penetration-testing and hacking-training firm—about cybersecurity-related career opportunities. In early 2016, the IEEE Computer Society partnered with NotSoSecure to develop cybersecurity eLearning courses for its members (www.computer.org/artofhacking).

ComputingEdge: What careers in cybersecurity will see the most growth in the next several years?

Haagman: Currently, there is a global shortage of cybersecurity skills in general. And demand for those capabilities is rising at an unprecedented rate. There are simply not enough people who can code securely or test code for technical vulnerabilities. Developers, by and large, lack coordination in the security methodologies they use, which is natural in any new field. There is a need for individuals who can help move the field forward quickly.

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

Haagman: I would say, “Don’t tell me you know how to do something. Show me.” Immerse yourself in hands-on applicable skills. This is critical, whether you work in the public or private sector. Build a lab, participate in knowledge sharing, collaborate. Academics, while not unimportant, are no substitute for experience.

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

Haagman: Be current. Know what is going on out there now. Also, developers and security testers must be able to sift quickly through data, analyze it effectively, and apply the resulting knowledge to produce an appropriate decision.

ComputingEdge: How can new hires make the strongest impression from the beginning?

Haagman: Show a hunger and desire to develop yourself professionally and to know your subject thoroughly.

ComputingEdge: Name one critical mistake young graduates should avoid when starting their careers?

2469-7087/17/$33.00 © 2017 IEEE

Haagman: I'll give you two. First, avoid not having sufficiently broad experience in your field. Immerse yourself in the field, and enjoy yourself. Second, never do anything illegal. Ever. It's wrong and also totally unnecessary. Respect the Internet and your career. It's a wonderful playground and opportunity, but remember that your name and integrity are incredibly important. So, never hack or do anything without permission or without having the right safety mechanisms in place. It's a fundamental moral issue.

ComputingEdge: Do you have any learning experiences you could share that could benefit those just beginning their careers?

Haagman: When I first started my career, I threw myself into every project I could get my hands on. I pursued every certification I could get, built labs, and read every book and website I could. The key is to make sure that your certifications are relevant to your field and your skill level, and that they help you advance. We are in the midst of a magical time—an extraordinary era in tech that we're unlikely to see again—that brings a significant number of opportunities to those who want a technology career. Make the most of it.

ComputingEdge's Lori Cameron interviewed Haagman 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 Haagman at dan@notsosecure.com.

CLOUDERA, INC. is recruiting for our Palo Alto, CA office: Software Engineer: Plan, design & implement functional, system & regression tests. Mail resume w/ job code #35998 to: Cloudera, Attn.: HR, 1001 Page Mill Rd., Bldg. 3, Palo Alto, CA 94304.

CLOUDERA, INC. is recruiting for our Palo Alto, CA office: Software Engineer: As a key member of the team, create & deliver our product stack deployment in Cloud environments. Mail resume w/ job code #37393 to: Cloudera, Attn.: HR, 1001 Page Mill Rd., Bldg. 3, Palo Alto, CA 94304.

EPIC HYPERSPACE DEPLOYMENT ADMINISTRATOR. Provide second tier support including troubleshooting, break/fix, software implementations, upgrades and maintenance as needed. Perform installation, administration and Operations and Maintenance (O&M) support for all aspects of Citrix infrastructure, including deployments, migrations and updates. Troubleshoot complex issues in a timely manner as necessary to maintain the performance and stability of the Citrix production infrastructure. Assist in creation of documented standard processes and procedures for all aspects of Citrix infrastructure, administration and management. Apply to: Gerald O'Mara, #82115, AHS Hospital Corp, 100 Madison Avenue, Morristown, NJ 07960.

SOFTWARE DEVELOPERS (3 positions) (Islandia, NY) Convert detailed systm dsgn & flow charts into Loan Origination S/ware Products using Java & J2EE following MVC architecture. Provide techn'l support by investigating & fixing defects. Communicate w/ customers to understand reqmts & provide instructions for operating personnel. Perform manual & automating testing w/ Junit & Selenium. Bachelor's in Comp Sci or Civil Engg (or foreign deg equiv) + 2 yrs exp req'd. Employer will accept master's deg in Comp Sci or Civ Engg (or foreign deg equiv) in lieu of this combo (BS + 2 yrs exp). Send res to Teledata Communications, Inc., 1377 Motor Pkwy, Ste 400, Islandia, NY 11749.

ASSOCIATE CONSULTANT/SYSTEMS ANALYSTS to design, develop, and test core architecture -sought by established IT firm. Qualified applicants will have a Master's or equiv. in Engineering (any field) and 12 mos' relevant industry exp.; or a Bachelor's or equiv. in Engineering (any field) and at least 5 yrs' progressively responsible relevant industry exp. Positions located in New York, NY & are subject to relocation to various unanticipated locations throughout the U.S. Mail resumes to: Tata Consultancy Services Limited, 9201 Corporate Blvd., Suite 320, Rockville, MD 20850 (Attn: A. Jindal).

PROGRAMMER ANALYST - design, develop, test & implement application s/w utilizing knowledge of Interactive/Web 2.0 technologies like HTML5, CSS3, JavaScript Scripting, JS framework, jQuery, NodeJS and AngularJS. 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.

TECHNOLOGY

Oracle America, Inc.

has openings for

SYSTEMS ANALYST - SUPPORT

positions in **Frisco, TX**.

Job duties include: Acts as an expert member of the problem-solving/avoidance team. Solves extremely complex (often previously unknown), critical customer issues.

Apply by e-mailing resume to sachin.u.shah@oracle.com, referencing 385.17749.

Oracle supports workforce diversity.

TECHNOLOGY

Oracle America, Inc.

has openings for

HARDWARE DEVELOPER

positions in **Burlington, MA.**

Job duties include: Evaluate reliability of materials, properties and techniques used in production; plan, design and develop electronic parts, components, integrated circuitry, mechanical systems, equipment and packaging, optical systems and/or DSP systems.

Apply by e-mailing resume to ray.bowden@oracle.com, referencing 385.17613.

Oracle supports workforce diversity.

TECHNOLOGY

Oracle America, Inc.

has openings for

HARDWARE DEVELOPER

positions in **Austin, TX**.

Job duties include: Evaluate reliability of materials, properties and techniques used in production.

Apply by e-mailing resume to greg.a.smith@oracle.com, referencing 385.18497.

Oracle supports workforce diversity.

COPILOT PROVIDER SUPPORT SERVICES, INC. in New Hyde Park, NY seeks Sr Web Dvlpr to design, code, & modify website layout, functions, apps & content, enhance existing sites, programs to ensure data processing productions sys continue to meet end user needs, dvlp appropriate code structures to test, maintain & implement pages, app & contingency plans, extend existing app through enhancement & upgrades to ensure sys's continue to meet co. req'ts, assist & support in routine maint., upkeep & dvlpmt of websites, research & propose softw programs compatible w/ co's goal & future dvlpmt, assume ownership of code thruout dvlpmt, staging, testing, productn & post-productn, prep & coord. intranet web updates & website changes. Must have Master's in Web Design or CS, plus 6 months exp in web design. Mail resume: Copilot Provider Support Services Inc., 1981 Marcus Ave, #C130, New Hyde Park, NY 11042 (Attn: Vanessa Mariacher).

SOFTWARE

Oracle America, Inc.

has openings for

SOFTWARE DEVELOPER

positions in **Culver City, California.**

Job duties include: Design, develop, troubleshoot and debug software programs for databases, applications, tools, and networks. Apply basic to intermediate knowledge of software architecture to perform software development tasks associated with developing, debugging or designing software applications or operating systems according to provided design specifications.

Apply by e-mailing resume to
david.chieu@oracle.com,
referencing 385.18069.
Oracle supports workforce diversity.

Cisco Systems, Inc.

is accepting resumes for the following positions:

BELLEVUE/SEATTLE WA: Technical Lead/Leader (Ref.# BEL11): Lead engineering groups on projects to design, develop or test hardware or software products.

BOXBOROUGH, MA: Customer Support Engineer (Ref.# BOX7): Responsible for providing technical support regarding the company's proprietary systems and software.

ISELIN/EDISON, NJ: Systems Engineer (Ref.# ED13): Provide business-level guidance to the account team or operation on technology trends and competitive threats, both at a technical and business level. Travel may be required to various unanticipated locations throughout the United States.

MOORESTOWN, NJ: Solutions Integration Architect (Ref.# MOO4): Perform analysis and diagnosis of highly complex networking problems and network designs. Travel may be required to various unanticipated locations throughout the United States.

RESEARCH TRIANGLE PARK, NC: Network Consulting Engineer (Ref.# RTP954): Responsible for the support and delivery of Advanced Services to company's major accounts. Telecommuting permitted. **Customer Support Engineer (Ref.# RTP302):** Responsible for providing technical support regarding the company's proprietary systems and software. Telecommuting permitted.

RICHARDSON, TX: Network Consulting Engineer (Ref.# RIC22): Responsible for the support and delivery of Advanced Services to company's major accounts. Travel may be required to various unanticipated locations throughout the United States. **Solutions Architect (Ref.# RIC114):** Responsible for IT advisory and technical consulting services development and delivery. Travel may be required to various unanticipated locations throughout the United States.

SAN FRANCISCO, CA: CNG Member of Technical Staff (Ref.# SF9): Design, implement, and test software for a web application used by our customers for IT management. **CNG Management (Ref.# SF53):** Architect and develop requirements documents for new products.

SAN JOSE/MILPITAS/SANTA CLARA, CA: Principal Engineer (Ref.# SJ674): Responsible for defining requirements, designing architecture and implementing and delivering various components. **IT Manager (Ref.# SJ184):** Drive improvements to the strategic, core and support processes of Company's Commerce using data and analytics. **Quality Engineer, Failure Analysis (Ref.# SJ864):** Perform in-depth failure analysis on circuit board components and assembly throughout different stages of the product including Design, Manufacturing and Field. **Hardware Engineer (Ref.# SJ557):** Participate on development of Application Specific Integrated Circuit (ASIC) for next generation data center switch product family, with emphasis on routing/switching protocols. **Hardware Engineer (Ref.# SJ5):** Responsible for the specification, design, development, test, enhancement, and sustaining of networking hardware.

PLEASE MAIL RESUMES WITH REFERENCE NUMBER TO CISCO SYSTEMS, INC., ATTN: V51B, 170 W. Tasman Drive, Mail Stop: SJC 5/1/4, San Jose, CA 95134. No phone calls please. Must be legally authorized to work in the U.S. without sponsorship. EOE.

www.cisco.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.

Symantec Corporation

currently has openings for the following positions in **Culver City, CA (various levels/types)**:

Engineering Managers (EMCC117) 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.

Program Managers (PMCC117) Work closely with engineering members, managers, and leads, product managers, ensure rapid execution and on time, high quality delivery of software projects.

Software Engineers (SWECC117) Responsible for analyzing, designing, debugging and/or modifying software; or evaluating, developing, modifying, and coding software programs to support programming needs.

Software QA Engineers (SQACC117) Responsible for developing, applying and maintaining quality standards for company products. Develop and execute software test plans. Analyze and write test standards and procedures.

Symantec Corporation currently has openings for the following positions in **Mountain View, CA (various levels/types):**

Computer Systems Analysts (CSAHQ117) 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.

Product Managers (PDMHQ117) 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 (PMHQ117) 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 (SWEHQ117) Responsible for analyzing, designing, debugging and/or modifying software; or evaluating, developing, modifying, and coding software programs to support programming needs.

Software QA Engineers (SQAHQ117) Responsible for developing, applying and maintaining quality standards for company products. Develop and execute software test plans. Analyze and write test standards and procedures.

Database Managers (DBMHQ117) Support all non-product and product databases including installation, configuration, upgrade, backup and recovery. Design, install, configure and maintain monitoring system.

Symantec Corporation currently has openings for the following positions in **San Francisco, CA (various levels/types)**:

Software Engineers (SWESF117) Responsible for analyzing, designing, debugging and/or modifying software; or evaluating, developing, modifying, and coding software programs to support programming needs.

Research Engineers (1648.2316) Identify, evaluate, and recommend syntactic, machine learning, and statistical methods and models for natural language processing, content classification, and image recognition. Consult and leverage the latest academic advances to keep Symantec in the leadership position in product innovation.

Program Managers (PMSF117) Work closely with engineering members, managers, and leads, product managers, ensure rapid execution and on time, high quality delivery of software projects. Some travel required between Mountain View, CA and San Francisco, CA.

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.

Symantec.

COMPUTER PROGRAMMER. Create, modify, & test the code, forms, & script that allow computer applications to run. Write computer programs to store, locate, & retrieve specific documents, data, & information. Correct errors by making appropriate changes & rechecking the program to ensure that the desired results are produced. Utilize C, C++, Python, Java, SQL, XML, Red Hat, LINUX, SQL Server. Will work in unanticipated locations. Req. 2 years of experience. Send resume to Saxon Global, Inc., Attn: HR, 1320 Greenway Dr. Ste. 660, Irving, TX 75038.

SOFTWARE ENGINEER. (Search/Booking), KAYAK Software Corporation (Concord, MA): Will design, dev, & implement srch & booking features for car & package bookings on company pltfm. Min reqs: MSc in CS, Soft Eng, or rel field. Position also requires dmnstrtd wkng knwldg of: Multi-thread soft dev w/ JAVA; Hive/Hadoop & SQL data process tech; & Service implementation w/ REST and JSON. Send cover letter & resume to talent@kayak.com w/ ref to code JB16.

PRINCIPAL SOFTWARE ENGINEER (BJSS, NY, NY) Serve as lead IT consultant to clients in the financial services sector leveraging advanced knowledge of IT systems designed to support financial trading, pricing & risk analysis. Bachelor's Degree (3 yr. degree acceptable) in comp. sci. or related followed by 6 yrs. of progressive exp. in position offered or related. At least 3 yrs. of exp. serving as a consultant advising clients within financial services industry on architecture, development & implementation of IT systems to support financial trading activities throughout trading lifecycle, providing technical leadership for project work. Full term of required exp. must involve: Utilizing agile (e.g. scrum, xp, etc) & test-driven development methodologies (e.g. Junit, TestNG, Mockito, etc.) to develop server-side, enterprise-wide, high availability throughput trading systems; Implementing continuous integration workflows utilizing software packages such as Hudson, Jenkins, Teamcity &/or Bamboo; Utilizing version control systems with support for branching such as Perforce, Git, SVN &/or Mercurial; Designing & developing distributed multitiered systems & relational databases incorporating service oriented architecture; & Configuring & implementing message based systems based on technologies such as TIBCO, Zero MQ, Rabbit MQ, or Informatica. Position requires travel up to 90% of time to various client sites. Apply by mail, referencing job code VG/13279 to Office Administrator, BJSS, 14 Wall St., Ste. 2069, New York, NY 10005.

SENIOR SOFTWARE ENGINEER (Applications) sought by Alarm Lock Systems, LLC. of Amityville, NY, in electronic & mechanical access & egress control keyless entry products incl h/ware devices, d/base, s/ware systms, for end-to-end from product concept, dsgn planning, modeling, prototyping, to product go-life. Min. req.: BSc. in Comp Sci, or Engg, IT, or rltd tech'l field, or foreign deg evaluated to be equiv to US BS deg in same, + 5-yr exp in specific skill-sets. Mail resume to Alison Walsh, HR Dir, Alarm Lock Systems LLC., 333 Bayview Ave, Amityville, NY 11701. EOE. No calls/walk-ins.

THE UNIVERSITY OF
ALABAMA IN HUNTSVILLE

The University of Alabama in Huntsville

The Department of Computer Science of The University of Alabama in Huntsville (UAH) invites applicants for a tenure-track faculty position at the Assistant Professor level beginning August 2017. The incumbent will augment the department's emphases in at least one of the following areas: cloud computing, particularly secure cloud computing; mobile computing, particularly secure mobile computing; or data science, particularly big data applications. Outstanding candidates who couple cybersecurity with other areas of computing could also be considered.

A Ph.D. in computer science or a closely related area is required. The successful candidate will have a strong academic background, perform funded research, be able to carry out research in areas typical for publication in well-regarded academic conference and journal venues, and be keen on undergraduate education.

The department has a strong commitment to excellence in teaching, research, and service; the hire should have good communication, strong teaching potential, and research accomplishments.

UAH is located in an expanding, high technology area, next door to one of the largest research parks in the nation. Nearby are the NASA Marshall Space Flight Center, the Army's Redstone Arsenal, and many high-tech industries. UAH also has an array of research centers, including in information technology, modeling and simulation, etc. In short, collaborative research opportunities are abundant, and many well-educated and highly technically skilled people are in the area. There is also access to excellent public schools and inexpensive housing.

UAH has approximately 8500 students. UAH Computer Science offers the BS, MS, and PhD degrees in Computer Science and the MS and PhD degrees in modeling and simulation. Approximately 550 undergraduate majors and 175 graduate students are associated with the unit. Faculty research interests are many and include cybersecurity, mobile computing, data science, software engineering, visualization, graphics and game computing, multimedia, AI, image processing, pattern recognition, and distributed systems. Recent NSF figures indicate the department ranks 30th in the nation in overall federal research funding.

Interested parties should submit a detailed resume with references to info@cs.uah.edu or Chair, Search Committee, Dept. of Computer Science The University of Alabama in Huntsville, Huntsville, AL 35899. Qualified female and minority candidates are encouraged to apply. Initial review of applicants will begin immediately and continue until a suitable candidate is found. UAH is an equal opportunity/affirmative action institution.

ALGORITHMIC DEVELOPER. Develop internal software to automate analysis of financial data using C/C++. Programmatically simulate trading, performance, and risk. Use STATA for data processing and analysis. Develop SQL for financial data storage and maintenance. Create software solutions with Application Programming Interfaces to introduce new features. Scripting/Automation with Python. Apply to: Town Square Trading LLC, Attn: AD16, 1 World Trade Center, Ste 45C, NY, NY 10007.

SENIOR LAB ENGINEER wanted in Madison Heights, Michigan to supervise, verify and administer tests for Ficosa mirrors, surge tanks and washer systems. Send resume to Manuela Marin, Commercial Department, Ficosa North America Corp., 30870 Stephenson Hwy., Madison Heights, MI 48071.

SR. SOFTWARE ENGINEER (iOS Developer), KAYAK Software Corp (Cambridge, MA): Dev & maintain native iOS App for functionality & user exp. Min reqs: Bachelor's in CS or rel field, plus 1 yr exp w/ sftwr eng or mobile dev. Must have dmnstrtd wrkng knwldg of: App dev using object oriented dsgn patterns; Consummation of remote web serv using RESTful APIs & JSON objects; & source control mgmt using GIT. Send cover letter & resume to talent@kayak.com w/ ref to KG16.

SR. SVCS CNSLTNT (NY, NY & unanticip client sites thrght US) Implmnt CA Sec Mgmnt prods. Anlyze cmplx cust reqs. Monitor CA's intrnl prog dvlpmnt, train clients on CA SSO sftwre & prvde tech suprt. Prvde pre- & post-sales tech suprt. Resvle cmplx probs. Dvlp trial systems for cust. REQS: Bach Deg or for equiv in Comp Sci, CIS/IS, Math, Engg (any) or rel + 5 yrs prog exp in job &/or rel occup. Must have exp w/CA Single Sign On; Archtctng & configrng fed prtnrshps; Configrng PWPs to cust reqs; CA Prod Supp Process; Freq travel to unanticip client sites thrght US; wrk fr home anywhere in US. Send resume to: Althea Wilson, CA Technologies, 201 North Franklin Street, Suite 2200, Tampa, FL, 33602, Refer to Requisition # 145311.

SVCS ARCHITECT (NY, NY & unantcptd client sites in US) Architect, design & implmnt solutions w/in a client envrnmnt. Design, program, script, scope, & deliver CA solutions. Implmnt open source Java EE app server JBOSS (2). Troubleshoot tech issues & assess client's infrstrctre, bus reqs & planned budget to design solutions. REQS: Bach Comp Sci, Math, Engg (any) or rel +5 yrs prog exp in job &/or rel occup. Must have exp w Archtcting & implmnting CA Single Sign On (SSO), CA Identity Manager (IM) & CA Secure Cloud solutions w/in client envrnmnt; Implmnting open source Java EE app server, JBoss; Prgrmming & scripting w Java, Java Script & Kettle Script; Freq travel to unantcpted client sites in US; Work from home anywhere in US. Send resume to: Althea Wilson, CA Technologies, 201 North Franklin Street, Suite 2200, Tampa, FL, 33602, Refer to Requisition #145446.

TECHNOLOGY

Oracle America, Inc.

has openings for

SOFTWARE DEVELOPER

positions in **Columbia, MD**.

Job duties include: Design, develop, troubleshoot and/or test/QA software.

Apply by e-mailing resume to joy.ganguly@oracle.com, referencing 385.18191.

Oracle supports workforce diversity.

SOFTWARE

Oracle America, Inc.

has openings for

SOFTWARE DEVELOPER

positions in **Westminster, CO**.

Job duties include: Design, develop, troubleshoot and/or test/QA software.

Apply by e-mailing resume to ross.levin@oracle.com, referencing 385.20033.

Oracle supports workforce diversity.

TECHNOLOGY

Oracle America, Inc.

has openings for

SYSTEMS ADMINISTRATORS

positions in **Jacksonville, FL**.

Job duties include: Create and implement system enhancements to improve the performance and reliability of Oracle Taleo Enterprise Cloud Service worldwide. May telecommute from home.

Apply by e-mailing resume to Richard.paquet@oracle.com, referencing 385.15789.

Oracle supports workforce diversity.

SVC ARCHTCT (NY, NY & unanticip client sites thr US) Archtct & implmnt CA Sec Prods. Prvde on-site & rmte asstnce fr the dsgn & implmntn of CA's Security Prtflio. Prvde key tech strategies & sec measurs & prvde feedback to Prod Brand dvlprs. Prvde tech training fr intrnl world-wide staff & bus prtnrs. REQS: Bach deg or for equiv in Comp Sci, Math, Engg (any) Bus Admin or rel + 5 yrs prog exp in job &/or rel occup. Must have exp w/ CA IdentityMinder & PolicyXpress; Multiprod archtctres using CA Single Sign On, CA IdentityMinder & CA Privileged Idnty Mgr; Archtctng & cnfgrng CA Directory; Cnfgrng Packaged Wrk Prods (PWPs) to cust reqs. Freq travel to unanticip client sites thr the US req. Wrk fr hme anywhere in the US. Send resume to: Althea Wilson, CA Technologies, 201 North Franklin Street, Suite 2200, Tampa, FL, 33602, Refer to Requisition #145445.

IT PROFESSIONALS. (Business) Systems Analysts, Functional Business Analysts, Programmer Analysts, Software Engineers, Senior Solution Architects, and Senior Software Engineers sought by Sagitec Solutions, LLC, an established global technology solutions company. (Business) Systems Analyst require Master's degree or equiv. in Comp. Sc., IT, Engg (any), Business or related and 12 mos' relevant indus. exp. (will also consider candidates with bachelor's or equiv. in the stated fields and 5 yrs progressive, relevant indus exp.); experience with Agile design methodology and SCRUM framework, and as a team lead is req'd. Functional Business Analyst require Master's degree or equiv. in Comp. Sc., IT, Engg (any), Business or related and 12 mos' relevant indus. exp. (will also consider candidates with bachelor's or equiv. in the stated fields and 5 yrs progressive, relevant indus exp.); experience as a team lead is req'd. Programmer Analyst require Bachelor's or equiv. in Comp. Sc., IT, Engg (any) or related and 12 mos' relevant indus. exp.; pension and retirement systems domain experience is req'd. Software Engineer require Bachelor's or equiv. in Comp. Sc., IT, Engg (any) or related and 24 mos' relevant indus. exp. Sr. Solution Architect require Master's degree or equiv. in Comp. Sc., IT, Engg (any), or related and 12 mos' relevant indus. exp. (will also consider candidates with bachelor's or equiv. in the stated fields and 5 yrs progressive, relevant indus exp.); hands-on experience in design, development and implementation of modules for large-scale business application systems, and experience with Microsoft SQL is req'd. Sr. Software Engineers require Master's or equiv. in Comp. Sc., IT, Engg (any) or related and 12 mos' relevant indus exp. (will also consider candidates with bachelor's or equiv. in the stated fields and 5 yrs progressive, relevant indus exp.). All positions based out of Sagitec HQ in Little Canada, MN and subject to reloc. to various unanticipated sites in U.S. Mail resumes to Sagitec Solutions, LLC, ATTN: Asst. Manager-HR, 422 County Road D. East, Little Canada, MN 55117.

JUNIPER
NETWORKS

Juniper Networks is recruiting for our Sunnyvale, CA office:

Software Engineer #40839: Design, develop troubleshoot and debug hardware-dependent software drivers and tools for system technologies.

Software Engineer #15403: Design, develop, troubleshoot and debug features and implement systems and solutions to automate the development, testing, and integration of tools and applications involved in Continuous Delivery.

Software Engineer #40534: Design, develop, troubleshoot and debug a distributed and scaled infrastructure for routing protocols on Company products.

Sr. Strategic Delivery Engineer #39246: Design, develop, and implement regression, scale, migration, certification and end-to-end solution testing efforts.

Services Management Sr. Manager #32201: Work closely with the customer and multiple internal organizations to oversee the effective implementation of Customer Support and solutions. Telecommuting allowed.

Software Engineer #35020: Develop carrier-grade software products from proof-of-concept through production and into sustaining. Participate in the design of software platform architecture.

Product Marketing Manager #33849: Work with the Public Relations and Social Media team to develop marketing plans and activities for Company product lines, including press releases, executive voice blogs, and online social media conversations.

Information Developer Engineer Staff #40562: Create, evaluate, develop and modify user assistance model prototypes to support Company's GUI applications.

Resident Engineer #40922: Develop and maintain expertise in Company products to support critical production network solutions. May work at other undetermined worksites throughout the U.S.

Software Engineer #33367: Design, develop, troubleshoot, support and debug routing and switching functional testing within the continuous delivery environment.

Strategic Delivery Manager Staff #10587: Design, develop and implement regression, scale, migration, certification and end to end solution testing efforts. Complete end-to-end test automation from design to execution.

ASIC Engineer #26133: Perform ASIC verification for large, complex high-speed ASICs for Juniper's next generation of networking products.

Mail single-sided resume with
job code # to
Juniper Networks
Attn: MS A.4.435
1133 Innovation Way
Sunnyvale, CA 94089

CLOUDERA, INC. is recruiting for our Palo Alto, CA office: Sales Engineer: develop prototype & Proof of Concept as a starting point for engineering. Travel Reqd. Mail resume w/job code #37144 to: Cloudera, Attn.: HR, 1001 Page Mill Rd., Bldg. 3, Palo Alto, CA 94304.

SOFTWARE ENGINEER. Develop, write, debug & implement code for assigned game software projects.REQS: BS in CS in Real-Time Interactive Simulation, or rel fld or FDE & coursework in specialized skills. Position at Nintendo Software Technology Corp, located in Redmond, WA. See https://www.worksourcewa.com & Job ID 179398661 for details & reqs. Apply to: jodial02@Nintendo.onmicrosoft.com & ref job #110000008Y.

GLITTERSOFT GROUP, an IT consulting services provider headquartered in Starkville, MS, is hiring Software Engineers, Senior Software Engineers, and Technical Team Leads. Software Engineers should hold a Master degree in a related field and have at least 6 months of professional experience. Senior Software Engineers and Technical Team Leads should hold a Master degree in a related field and have at least 6 months of professional experience, or hold a Bachelor degree in a related field and have at least 5 years of progressively responsible professional experience. Necessity to relocate to various unanticipated worksite locations throughout the U.S. possible. All interested and qualified candidates should send their resumes to Madhu via email at madhu@glittersgroup.com or mail them to 60 Technology Blvd., Starkville, MS 39579. Please reference Job ID# 411965 for Software Engineer position, Job ID# 794905 for Senior Software Engineer position, and Job ID# 083201 for Technical Team Lead position.

SOLUTION DESIGNER. Travelers has openings in Hartford, CT for Solution Designers. Accountable for the devl., automation, compilation, & report preparation, i/c collecting data & profiles as needed, the eval. & analysis of data, the integ. of data, the devl. (prototyping & prod. build), & unit test. Manages resources/budget for mult. projects & aligns projects to priorities. Accountable for sol. design satisfying bus. rqmts. Assists in the devl. of the strategic plan for Info. Del. Provides leadership devl. to staff. Supv. S/W Engineers on projects as needed. Must possess at least a master's degree or its equiv. in MIS, Mathematics, Finance, Statistics, Elect. Eng., CS, Comp. Eng. or rltd. fld. & at least 3 yrs of work exp. in Info. Del. or a rltd. fld. In the alternative, at least a bachelor's degree or its equiv. in MIS, Mathematics, Finance, Statistics, Elect. Eng., CS, Comp. Eng. or rltd. fld. & at least 5 yrs of prog. work exp. in Info. Del. or a rltd. fld. would be acceptable. Must possess exp. with the following: working w/ analytic tools/models; using SQL against mult. data sources; Working w/ Bus. Int.; Info. Del. practices & processes; programming languages i/c SQL and VBA; & exp. leading a team and/or managing others.

DATA ENGINEER - HARLAND CLARKE CORP. has an opening for the position of Data Engineer in San Antonio, TX to monitor server & database process health to ensure availability & response. To apply mail resume to Harland Clarke Corp, Attn: Monica 15955 La Cantera Pkwy, San Antonio, TX 78256 & refer to 16-TX6174.83.

TECHNOLOGY

Oracle America, Inc.

has openings for

HARDWARE DEVELOPER

positions in **Austin, TX**.

Job duties include: Evaluate reliability of materials, properties and techniques used in production; plan, design and develop electronic parts, components, integrated circuitry, mechanical systems, equipment and packaging, optical systems and/or DSP systems.

Apply by e-mailing resume to
rush.mehta@oracle.com,
referencing 385.18015.
Oracle supports workforce diversity.

TECHNOLOGY

Oracle America, Inc.

has openings for

APPLICATIONS DEVELOPER

positions in **Bedford, MA**.

Job duties include: Analyze, design, develop, troubleshoot and debug software programs for commercial or end-user applications. Write code, complete programming and perform testing and debugging of applications.

Apply by e-mailing resume to
ramakrishna.rao@oracle.com,
referencing 385.19740.
Oracle supports workforce diversity.

TECHNOLOGY

Oracle America, Inc.

has openings for

APPLICATIONS DEVELOPER

positions in **Irving, TX**.

Job duties include: Analyze, design, develop, troubleshoot and debug software programs for commercial or end-user applications.

Apply by e-mailing resume to
kevin.burns@oracle.com,
referencing 385.16392.

Oracle supports workforce diversity.

ERICSSON INC. has opening for the following positions: PROJECT MANAGER_ Ericsson Inc. has openings in **BELLEVUE, WA** for scheduling, tracking, & implementing projects supporting key customer deliverables to the highest customer satisfaction, while driving Cost, Quality, and Timeliness. To apply mail resume to Ericsson Inc. 6300 Legacy Dr, R1-C12, Plano, TX 75024 & reference **Job ID# 16-WA-2659**.

ENGINEER - SERVICES SOFTWARE _ ERICSSON INC. has openings in **BELLEVUE, WA** to analyze, prepare, implement & verify the configuration & integration of a node, network or system. To apply mail resume to Ericsson Inc. 6300 Legacy Dr, R1-C12 Plano, TX 75024 & reference **Job ID# 16-WA-2721**.

ENGINEER – SOFTWARE _ ERICSSON INC. has openings in **EL SEGUNDO, CA** to engage with product line support and maintenance to troubleshoot production issues. Up to 50% domestic and/or international travel required. To apply please mail resume to Ericsson Inc. 6300 Legacy Drive, R1-C12 Plano, TX 75024 & reference **Job ID# 16-CA- 2633**.

ENGINEER – SERVICES SOFTWARE _ ERICSSON INC. has openings in **PLANO, TX** to participate in software loading, configuration, integration, verification, & troubleshooting of solutions. Frequent travel required. To apply mail resume to Ericsson Inc. 6300 Legacy Dr., R1-C12, Plano, TX 75024 & reference **Job ID# 16-TX- 3498**.

ENGINEER – SERVICES SOFTWARE _ ERICSSON INC. has openings in **PLANO, TX** to perform network analysis, planning, syst design, & network performance audits related to Core Network Competence. Up to 35% domestic and/or international travel required. To apply mail resume to Ericsson Inc. 6300 Legacy Dr., R1-C12, Plano, TX 75024 & reference **Job ID# 16-TX- 3515**.

SR SOFTWARE ENGINEER (IOS DEVELOPER), KAYAK SOFTWARE CORP (CAMBRIDGE, MA): DEV & MAINTAIN NATIVE IOS APP FOR FUNCTIONALITY & USER EXP. Min reqs: Bachelor's in CS or rel field, plus 1 yr exp w/ sftwr eng or mobile dev. Must have dmnstrtd wrkng knwldg of: App dev using object oriented dsgn patterns; Consummation of remote web serv using RESTful APIs & JSON objects; & source control mgmt using GIT. Send cover letter & resume to talent@kayak.com w/ ref to KG16.

AD NETWORK DEVELOPER, KAYAK Software Corporation (Cambridge, MA): Will des, dev, & maintain sftwr apps supporting ad netwrk. Min reqs: Master's in CS, Comp Eng, or rel, plus 1 years' exp w/ sftwr eng. Must also have wrkng knwldg of: 1) SQL, NoSQL, and Hadoop data proc. tech; & 2) Lifecycle of REST API dev. Send cover letter & resume to talent@kayak.com w/ ref to code UP17.

SENIOR SYSTEMS ANALYST, Chandler, AZ: Limited domestic travel and/or occasional relocation to multiple client locations nationwide to define IT architecture/integration strategies using Oracle, Java based technologies. Coordinate team of developers. Review and analyze business processes and map them to functionality provided by Oracle products. Work in multiplatform environment. Reply to: Pravici, LLC, 3115 S. Price Rd., Suite #132, Chandler, AZ 85248.

TECHNOLOGY

Intuit Inc.

has openings for the following positions in **Mountain View, California**:

Senior Technical Data Analysts (Job code: I-2614): Work directly with product developers, analysts and marketers to recommend and implement data tracking and reporting to answer business questions. **Staff Software Engineers in Quality (Job code: I-1734):** Apply mastery of software engineering to design, influence and drive Quality and testability of products and services. **Application Operations Engineers (Job code: I-2850):** Exercise judgment within best business operations practices to design, implement, and support operational standards and capabilities for individual software products or connected services. **Senior Product Managers (Job code: I-322):** Work with Finance business process owners and relevant Finance stakeholders to translate business requirements to technology solutions related to Billing, Revenue and Payment Applications. **Staff Software Engineers in Quality (Job code: I-187):** Apply master level software engineering and industry best practices to design, implement, and support software products and services.

Positions in **San Diego, California**:

Software Engineers (Job code: SW117-SD): 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. **Senior Software Engineers in Quality (Job code: I-205):** Apply senior level software engineering practices and procedures to design, influence, and drive quality and testability of products and services. **Staff Network Engineers (Job code: I-1790):** Design and implement new network technologies and architecture in support of our on-premise, hybrid and cloud environments.

Positions in **Plano, Texas**:

Software Engineers in Quality (Job code: I-2417): 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.

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.

SOFTWARE

Oracle America, Inc.

has openings for

SOFTWARE DEVELOPER

positions in **Seattle, WA.**

Job duties include: Design, develop, troubleshoot and/or test/QA software. As a member of the software engineering division, apply knowledge of software architecture to perform tasks associated with developing, debugging, or designing software applications or operating systems according to provided design specifications.

Apply by e-mailing resume to
lee.goodrich@oracle.com,
referencing 385.19971.
Oracle supports workforce diversity.

TECHNOLOGY

HomeAway.com, Inc

has openings for

Staff BI Engineers

(Job ID#: 728.4137)

in **Austin, Texas:**

Build and maintain a self-service layer that will enable all employees easy access for our core data elements.

To apply, send resume to: HomeAway/Expedia Recruiting, 333 108th Avenue NE, Bellevue, WA 98004. Must reference Job ID# .

TECHNICAL

Oracle America, Inc.

has openings for

TECHNICAL ANALYST

positions in
Colorado Springs, Colorado.

Job duties include: serving as the technical interface to customers, Original Equipment Manufacturers (OEMs) and Value-Added Resellers (VARs) for resolution of problems related to the installation, recommended maintenance and use of Oracle products.

Apply by e-mailing resume to
brad.ericksen@oracle.com,
referencing 385.19775.
Oracle supports workforce diversity.

TECHNICAL

Oracle America, Inc.

has openings for

TECHNICAL ANALYST

positions in **Orlando, FL.**

Job duties include: Deliver solutions to the Oracle customer base while serving as an advocate for customer needs.

Apply by e-mailing resume to
stefano.montero@oracle.com,
referencing 385.19834.

Oracle supports workforce diversity.

SOFTWARE

Oracle America, Inc.

has openings for

SOFTWARE DEVELOPER

positions in **Naples, FL.**

Job duties include: Design, develop, troubleshoot and/or test/QA software.

Apply by e-mailing resume to
Stefan.kursawe@oracle.com,
referencing 385.18182.

Oracle supports workforce diversity.

TECHNICAL

Oracle America, Inc.

has openings for

TECHNICAL ANALYST

positions in **Lehi, Utah**.

Job duties include: Deliver solutions to the Oracle customer base while serving as an advocate for customer needs.

Apply by e-mailing resume to
keith.tucker@oracle.com,
referencing 385.19867.

Oracle supports workforce diversity.

DATABASE ADMINISTRATOR (Disaster Recovery). Des./build/test/implement/administer databases, disaster recovery systems, using Oracle technology, write Unix scripts for database & app monitoring. U.S. Bach or foreign equiv. (Engineering) req'd. 5 yrs. prog. responsible exp. in database field req'd. Must have 3 yrs' exp. in pos'n(s) w/ a) design & build of disaster recovery systems using Oracle database technology & b) writing Unix shell scripts for database & app monitoring. STATS LLC, Chicago, IL. Resumes to: Recruiting, STATS LLC, 203 North LaSalle St, 22nd floor, Chicago, IL 60601.

SOFTWARE DEVELOPER IN TEST: Peterson Technology Partners Inc. seeks qualified Software Developer in Test for its headquarters located in Park Ridge, IL & various & unanticipated work locations thruout the U.S. Resp. for developing test scripts using SOA tools incl. estimating required resources & components for SOA testing, following standard testing methods. Master's degree in Comp Sci, Info System Tech, Electronics Engg, or a closely related field of study required (will accept Bachelor's degree in above fields + 5 yrs related progressive exp in lieu of Master's degree) w/ at least 2 yrs exp in: (i) developing & executing test scenarios, test scripts, test data docs based on design & test docs in Agile environ. as well as testing web services; (ii) ensuring that root cause analysis defects are done, & coordinating deployments to Quality Assurance (QA) & Production environs; & (iii) utilizing Selenium Web Driver & QTP to automate web apps. An EOE. Respond by mail to Peterson Tech Partners, 1030 W Higgins Rd. Ste 230, Park Ridge, IL 60068. Refer to ad code: PTP-0117.

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
Email: dsims@computer.org

TECHNOLOGY

Expedia, Inc.

has openings for the following positions in **Bellevue, Washington**:

Directors, Technology (Job ID#: 728.509): Manage group of database developers, application engineers, and operations resources to support Data Warehouse, Email Marketing, and Loyalty Operations. **Network Engineers (Job ID#: 728.1783):** Assist application and system owners in troubleshooting problematic network dependent applications. **Reporting and Analysis Managers (Job ID#: 728.2373):** Support, influence, and challenge business decisions with data and analyses. **Database Administrators (Job ID#: 728.2220):** Responsible for all phases of database administration such as installing, configuring, monitoring, troubleshooting, and maintaining SQL and NoSQL databases. **Directors, Product Analytics (Job ID#: 728.1026):** Develop, maintain, and improve tools and processes to track and report trends. Provide analytical tools that determine action plans and insight. **Oracle BI Developers (Job ID#: 728.2420):** Analyze and develop OBIEE 11G technical solutions. **Data Scientists (Job ID#: 728.1722):** Utilize data science methodologies, including forecasting, clustering, and classification. **Analytics Managers (Job ID#: 728.1872):** Deliver analysis support and data-driven guidance to the corporate leadership and internal clients.

To apply, send resume to: Expedia Recruiting, 333 108th Avenue NE, Bellevue, WA 98004. Must reference Job ID#.

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

Software Development Engineer (Multiple Positions Open) (Req# A5YPLB) Rsrch, des, dev, implmnt & debug stat & deterministic Natural Lang Processing SW as part of novel text input & text procssng sys.

Software Development Engineer (Req# 9BN3EE) Dsgn & dev SW for enabling Near Field Communication (NFC) & Apple Pay on Apple sys.

Software Development Engineer (Req#9GAP7B) Rsrch, des, dev, implmt, & debug compilers targeting current & future GPUs.

ASIC Design Manager (REQ#-9FLVN2) Wrk w/ eng teams to define new prod feats & enhancmnts.

Software Development Engineer (Req# A83VTC) Dsgn & dvlp SW systems for analyzing large-scale data & extracting insight from them.

Software Engineer Applications (Req# 9VUVSP) Build sys for lrg-scale data sci apps.

Software Engineer Applications Manager (Req# 9QS3DM) Mnge team of engrs working on iOS & Mac OS apps & libraries for enterprise users & customers.

Software Development Engineer (Req# 9RLR3K) Des & dev SW arch for large scale, multi-tier Apple Product Ops.

Product Design Engineer (Req# 9E8T9U) Support design of all PCB & Flex Circuits used in Apple prod. Travel Req 30%.

Software QA Engineer (Req# 9TLTPT) Dsgn investigtns & test power to drive sys quality across all Mac prods.

Software Development Engineer (Req# 9SYRKF) Des & dev OS lvl Networking SW, in prticlr Rem. Access VPN prtcls, TCP/IP config prtcls, 802.1x & CaptiveNetwork prtcls, acrss Apple's range of products.

Software Engineer Applications (Req# A8LVPM) Des & dev front-end web SW for maps eval.

Failure Analysis Engineer (Req# 9M93DA) Identify & drive imprvmnts to current & new prdcts by analyzing iPhone sys failure.

Software Development Engineer (Req# A9D23X) Des & dev SW for big data processing systems.

ASIC Design Engineer (Req# 9M522Q) Dev silicon tstng des for validating internal stndrd cell libry elements used in des of Apple prod's like iPhone, iPad, & iWatch.

Engineering Project Lead (Req# A3HM5U) Supp & enhance developer & customer rel App Store Ops.

Software Development Engineer (Req# A6M46V) Des & dev quark as a versiond data store for maps data in the contxt of the neutrn pltform.

Software Development Engineer (Req# A79PQ9) Des, dev, debug & test network device drivers, network protocol stack SW, IPCs & apps.

Software Engineer Applications (Req# 9VSSSA) Design and develop internal software tools.

Systems Design Engineer (Req# 9QLVDY) Eval Active OTA Perfrmnce. Characterize passive/active antenna perfrmnce incl efficiency, gain, & pattern.

Hardware Development Engineer (Req# 9WRQYD) Dsgn & dev displays for handheld devices. Travel req'd 15%.

Software Engineer Applications (Req# A4FQFT) Research, design, & implmnt Digital Security solutions.

Systems Design Engineer (Req# 9G92MT) Dsgn, dvlp, & test EMC solutions for IT and mobile communications equipment.

Engineering Manager (Req# 9GX296) Provide org & tech leadership for core compiler features, testing and releases.

Hardware Development Engineer (Req# 9ZLMXE) Dev novel battry cell sol for new & exstng Apple prods. Travel req 30%.

Software Engineer Applications (Req# AE932D) Provde dtbs archtctre & des solts for admnstrtng lrg dtbse infrstrctre.

ASIC Design Engineer (Req# 9TEU42) Conduct perfom tuning, correlation, & veri for low-pwr high-perform microprcss'r sys.

Software Development Engineer (Req# 9ZD4S7) Research, des, dev & test OS storage subsys & sys firmware that support various storage-related tech. Travel req 15%.

Software Development Engineer (Req# 9CKR5P) Des & dev SW for Search Infrastruct Internationalization. Language req: Spanish, Italian, or Dutch.

Systems Design Engineer (Req# 9TLRLH) Test OTA RF sensitivity perf across multiple radio tech. Travel req 15%.

Software Development Engineer (Req# 9WG2KX) Provide QA testing for help content delivered on the web or in-app on iOS, OS X, & other OS.

Engineering Project Coordinator (Req# 9YX3QH) Responsible for the product dvlpmnt methods for product safety compliance.

Software Engineer Applications (Req# 9SYRGU) Des, dev & deploy high-vol scalable server-side apps in Java/C/Lua.

Systems Design Engineer (Req# 9E5RMR) Dev & optimize RF autmtn sys for Apple's newest prods includ. iPhones, iPods, iPads & others using chipset-lvl calibrtn. Travel req'd 20%.

Product Design Engineer (Req# 9YAUYD) Dev & implmnt des for manufctrng prcesses & mthds for consmr prdcts. Travel req 20%.

Software Engineer Systems (Req# 9WQQ3T) Des and dev WiFi and Bluetooth Coexistence.

Software Development Engineer (Req# A3WNE6) Respnsble for tstng & validtn of pre-release SW.

Systems Design Engineer (Req# 9KTT27) Use Spanish & Portuguese to ensure eng dsgns follow regional regulation reqs for Latin American region. Travel req 25%.

Software Development Engineer (Req# 9T8VDK) Des & dev SW for GPU dev tools.

Systems Design Engineer (Req# 9QXRYG) Dsgn custom test instruments & test the electrical systms performance of iOS devices. Travel req. 20 %.

Software Engineer Applications (Req# 9Z6MFY) Dsgn & dvlp SW apps for fnctnl enhncmnts.

Software Engineer Applications (Req# 9ZPV6D) Bld websites & apps using Adobe Exp Mgr.

Systems Design Engineer (Req# 9HK33S) Dvlp & optmz RF autmtn syss used on Apple's newst prdcts incldng iPhones, iPods, iPads & others. Trvl req 25%.

Product Quality Engineer (Req# A8HSN2) Dvlp & implmnt quality systems for the Power products. Travel req. 25%.

Software Development Engineer (Req# A7Z3F6) Des & dev Bluetth specfctns, SW, & drivrs for dvcs & accesris.

Software Development Engineer (Req# A7U2UV) Des, dev & optimize GPU drivers for Apple HW products.

Software Quality Assurance Engineer (Req# 9M94KJ) Create and document test plans & test cases, along w/ strategy for execution in a short cycle.

Software Development Engineer (Req# ACUPVL) Build a routing pltfrm to deliver next gen of Maps srvcs.

Software Systems Engineer (Req# ACJTQQ) Des & dev SW for user apps & internal sys.

Software Engineer Applications (Req# A5H4JH) Archtct, dev & deploy hi-vol, mult-tiered, distrb'd mission critical apps.

Software Development Engineer (Req# A4K25K) Des, dev & execute auto tests for power & performance regressions, HW & oper sys.

Software Development Engineer (Req# A4RRX2) Des, dev, test, & maintain sw for internet advertising systems.

Machine Learning Engineer (Req# 9QEQHM) Dsgn & dev SW & machine learned sys for natrl language proc (NLP).

Engineering Project Coordinator (Req# 9WN2GJ) Coordinate lrg cross functional iOS & OS X SW proj.

Software Development Engineer (Req# A3R2RE) Des, dev, & help support the massively scalable data back-end for Siri.

Software Engineer Applications (Req# A5239S) Dev, create, impl, & support the web app devpmt of Sales Training App using large scale & high performing, obj oriented internet tech.

Software Development Engineer (Req# 9QK3V5) Dlvr high quality prdct releases w cellular prtcl tst & dvlpmnt enginrs.

Network Engineer (Req# A5FV8A) Responsible for the dev, delivery & ops of Apple's global VPN infrastructure. Travel req 20%.

Software Engineer Applications (Req# 9ZHQT8) Dsgn & implmnt entrprse level bck end solutions.

Systems Design Engineer (Req# A4Z3X2) Test OTA RF sensitivity performance across multiple radio tech. Travel req 15%.

Hardware Development Engineer (Req# 9H8QDP) Dev new transducer materials & tech to deliver world class portable audio products. Travel req 25%.

Mechanical Quality Engineer (Req# 9CB2EU) Contrib to dsgn of future Apple prod from a qual side. Travel Req'd 30%.

Continued on next page...

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

Operations Engineering Project Specialist (Req# 9Q4UKF) Dev & implmnt LCD prod tech & high vol manufctrng proc'sses for display. Trav Req 20%.

Supply Demand Planner (Req# A2E55J) Des, dev, test & eval projects that will support & execute Apple's refurbishment model.

Software Engineer Applications (Req# AB5W48) Des, dev, & maintain SW & tools for lrg-scale sys ops & deployment automation.

Software Engineer Applications (Req# 9LN55Y) Dsgn & implmnt e-commerce payment instruments for the Apple Online Store.

Software Engineer Applications (Req# A9GTSM) Dev highly sclable, reliable SW based on MicroServices and Service Orientd Arhtctre.

Software Development Engineer (Req# A54R22) Des & dev kernel SW, sys SW, & tls for performance analysis.

Operations Engineering Program Lead (Req# A4238R) Idntfy & exe optimizations of the iPhone mfg process.

Software Development Engineer (Req# 9FG2B9) Des & dev SW for Camera sys.

ASIC Design Engineer (Req# 9QLUDQ) Dev phys des methodology for the CPU of iPhone and iPad SOC.

Software Engineer Applications (Req# 9UNVWJ) Dvlp art camera algorithms fr mobile imaging devices, starting frm research & prototyping & delivering all the way thrgh prdction code.

Software Engineer Applications (Req# A87TD4) Dsgn, dev & spprt hi prfrmnce enterprise Hadoop solutions.

Hardware Development Engineer (Req# A57N62) Resp for HW & firmware dev for health sensing purposes. Travel req. 20%.

Engineering Program Lead (Req# AB7QDX) Coord prgrms for SW dev & ops for analytics-as-a-srvc infrstrctre team.

Software Development Engineer (Req# 9TEU6V) Res for screening incoming bugs, deciding actionable steps, & escalating to approp eng.

Software Development Engineer (Req# 9ZTVYC) Dev Siri's next gen speech recognition sys across dozens of lang & domains. Foreign language not req'd

Software Development Manager (Req# 9HRV32) Lead dvlpmnt team for sys dplymnt tools, config mgmt, sys montrng, & lg-scale distrb sys arch, eng, & implmnt.

Software Engineer Applications (Req# AANW4H) Rspnsbl for setup of Hadoop clusters w/ optimum configs.

Human Factors Design Engineer (Req# 9JLPEX) Condct user research studies for phys/digi design. Travel req: 25%.

Hardware Development Engineer [Multiple Positions] (Req# 9EZ396) Des, dev, debug & validate iPhone HW. Travel req: 20%.

Software Development Engineer (Req# 9VCS63) Qualfy latest Apple prdcts w/ a focus on storage & file systms.

Software Development Engineer (Req# 9WZ33E) Resp for des & dev real-time embed SW for telecomm systms reltd to baseband cell protcol stack SW.

Hardware Development Engineer (Req# 9SY3PK) Des, dev & test powr convrtrs incldng AC/DC, DC/DC & DC/AC.

Hardware Development Engineer (Req# A4YQGK) Des & dev the powr supply for Apple prdcts.

Hardware Development Engineer (Req# 9FKQ4U) Create new & novl lens des for imgng apps in moble dvcs. Travel req: 25%.

Lead Software Development Engineer (Req# 9HAULX) Dvlp predictive feats on the Apple Maps Pltfrm.

Firmware Engineer (Req# ABMT2Q) Dsgn, devel & debug firmware in power sys for Apple products. Travel req'd 20%.

Hardware Development Engineer (Req# ACZ22Z) Resp for sys-lvl analysis, dsgn & dev of new camera features & tech for perfrmnce enhncmnt.

Software Engineer Systems (Req# 9GW3ZW) Bld, specify, des, dev, & launch Apple's sensing tech prod characterization & prod instrumentation SW. Travel req: 25%

Hardware Development Engineer (Req# AAJ32M) Create des for cmplx sys, components & subassemblies. Travel req: 15%

Hardware Development Engineer (Req# A8C2KE) Drive the des, dev, integrtn of speakr, receivr, & microphon modls into Apple prdcts. Travel req 15%.

Software Engineering Manager (Req# 9TVR5D) Mng GPU SW Eng & lead dev of GPU drivers for OpenGL ES & Metal graph APIs on upcoming GPU architectures.

Firmware Engineer (Req# 9EZQA7) Des & dev firmware & SW for embdded accessories.

Software Development Engineer (Req# A8627D) Dsgn & dev apps SW for Android & other mobile devcs.

Software Engineer Applications

(Req# 9JG2HM) Dsgn & dev SW for Apple News ecsystm.

Systems Design Engineer (Req# A8X2XF) Des, dev, validate & oversee ongoing factory tests. Travel req'd: 20%.

Data Infrastructure Engineer (Req# 9TG3K5) Write & mntn SW for the purp of ingest, transform, & enrich mass data sets.

Operations Engineer Program Lead (Req# 9XXQC5) Des, dev & support new product introductions. Travel req 30%.

Hardware Development Engineer (Req# 9FCRKL) Dev & validate motion & enviro sensors for Apple mobile products. Travel req 15%.

ASIC Design Engineer (Req# 9V62L8) Dsgn & dev SW & HW for semiconductor debug, charactrztn & prodctn.

ASIC Design Engineer (Req# 9WASK2) Dsgn & dev SW & HW for semiconductor debug, chrctrzatn & prodctn.

Software Engineer Applications (Req# A7T2DV) Anlyze, des, code, inspct, debug & tst new SW sltns in the intrnl tools area with emphs on iOS/Mac applctn devlpmnt.

Software Engineer Applications (Req# 9XQSZK) Dev a netwrk operatng sys for commoditzd netwrk HW.

ASIC Design Engineer (Req# A4VVC9) Prepr & prfrm silcon valdatn of low-powr Cntrl Prcssng Units (CPUs) usd in mobl dvcs.

Software Engineer Systems (Req# 9WN2EM) Des, dev & supprt high perfrmnce entrprse Hadoop solutns.

Software Engineer Applications (Req# 9E635Z) Dev the next generation of cloud sup for Apple Oper Sys.

Hardware Development Engineer (Req# 9GC46G) Des & dev SW automton tools for specfc acoustc testng in telecomunctn & audio systms. Travel req 15%.

IST Technical Project Specialist (Req# A6J3GH) Impl global solutions for Apple Online & Retail store rel projects.

Systems Design Engineer (Req# 9DH2DV) Des & dev RF calibration & test algorithms for telecomm sys. Travel req 30%.

Producer (Database Marketing Engineer) (Req# AFYNJN) Wrk with teams in Dsgn, Prdctn, & Mrktng to prvde tech implmntn of database mrktng campaigns.

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

Unix/Linux Systems Engineer (Req# 9ZX3S7) Install & coordinate Apple's manuf critical sys.

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

Data Architect (Req# AEFNB8) Drive process improvem, sys enhance, create data analytical models, set ops benchmarks, provide detailed reporting & perform gen business intel duties.

Software Engineer Systems (Req# A3Z386) Oversee test automtn & release mgmt to help Analytic Insight team mitigate fraud, waste & abuse cmpny-wide.

Apple Inc. has the following job opportunities in Maiden, NC:

Information Systems Engineer (Req# A6F2VC) Admin, install, config, trblshoot & wrt supp doc of IBM/AIX HW tech.

Refer to Req# & mail resume to Apple Inc., ATTN: D.W.,
1 Infinite Loop 104-1GM,
Cupertino, CA 95014.

Apple is an EOE/AA m/f/disability/vets.

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

Industrial Designer (Req# 9TNQH6) Dev high-quality dsgn concepts to drive industry dsgn for new Apple prods. Travel req'd 20%.

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: D.W.,
1 Infinite Loop 104-1GM,
Cupertino, CA 95014.

Apple is an EOE/AA m/f/disability/vets.

Recognizing Excellence in High Performance Computing

Nominations are Solicited for the

SEYMOUR CRAY, SIDNEY FERNBACH, & KEN KENNEDY AWARDS

SEYMOUR CRAY COMPUTER ENGINEERING AWARD

Established in late 1997 in memory of Seymour Cray, the Seymour Cray Award is awarded to recognize innovative contributions to high performance computing systems that best exemplify the creative spirit demonstrated by Seymour Cray. The award consists of a crystal memento and honorarium of US$10,000. This award requires 3 endorsements.

Deadline: 1 July 2017

All nomination details available at

http://awards.computer.org

SIDNEY FERNBACH MEMORIAL AWARD

Established in 1992 by the Board of Governors of the IEEE Computer Society. It honors the memory of the late Dr. Sidney Fernbach, one of the pioneers on the development and application of high performance computers for the solution of large computational problems. The award, which consists of a certificate and a US$2,000 honorarium, is presented annually to an individual for "an outstanding contribution in the application of high performance computers using innovative approaches." This award requires 3 endorsements.

ACM/IEEE-CS KEN KENNEDY AWARD

Established in memory of Ken Kennedy, the founder of Rice University's nationally ranked computer science program and one of the world's foremost experts on high-performance computing. A certificate and US$5,000 honorarium are awarded jointly by the ACM and the IEEE Computer Society for outstanding contributions to programmability or productivity in high performance computing together with significant community service or mentoring contributions. This award requires 2 endorsements.

IEEE

IEEE computer society

acm Association for Computing Machinery

TECHNOLOGY

LinkedIn Corp.

has openings in our **Sunnyvale, CA** location for **Software Engineer (All Levels/Types) (SWE0117SV)** Design, develop & integrate cutting-edge software technologies; **Operations Engineer (6597.1449)** Monitor & resolve application, system, & network incidents affecting the company platform & ensure maximum availability. **Test Engineer (6597.896)** Design & develop advanced test suites using object-oriented methodologies. **Data Scientist (6597.1538)** Design & analyze experiments to test new product ideas & convert the results into actionable product recommendations. **Sr. Database Engineer (6597.1769)** Design, develop & integrate cutting-edge software technologies. **User Experience Designer (6597.1294)** Design solutions that address business, brand & user requirements. **Staff Network Engineer – Security (6597.1531)** Plan, deploy, and manage network security solutions. **Staff Site Reliability Engineer (6597.1310)** Apply the principles and techniques of Computer and Information Science to ensure that complex, web-scale systems are healthy, monitored, automated, and designed to scale. **Senior Site Reliability Engineer (6597.1759)** Serve as a primary point responsible for the overall health, performance, and capacity of one or more internet-facing services. **Senior Manager, Software Engineering (6597.1809)** Hire world class talent & provide technical guidance, career development, & mentoring to team members. **Performance Engineer (Software Engineer) (6597.1751)** Conduct performance analysis & code optimization across multi-tier & multi-data centers. **Senior Information Security Engineer (6597.842)** Responsible for protecting LinkedIn's infrastructure, applications, & members by identifying new vulnerabilities & responding to existing vulnerabilities within the organization. **Site Reliability Engineer (6597.1465)** Serve as a primary point responsible for the overall health & performance of one or more large-scale system. **Test Engineer (Software Engineer) (6597.1594)** Design, develop & integrate cutting-edge software technologies. **Data Scientist (6597.1523)** Extract & analyze data to drive product strategy. **User Experience Designer (6597.1665)** Collaborate with product managers to define the interaction design of products & visualize new concepts.

LinkedIn Corp. has openings in our **San Francisco, CA** location for **Software Engineer (All Levels/Types) (SWE0117SF)** Design, develop & integrate cutting-edge software technologies.

LinkedIn Corp. has openings in our **Calabasas, CA** location for **Manager, Database Engineering (6597.1339)** Leverage data architecture & warehousing skills to build a leading edge enterprise data warehouse encompassing the entire life cycle, including data integration, transformation, logical & physical design, security, backup, & archival strategies implementing industry best practices.

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

C 100