EDITORIAL

Musings

RIK FARROW

Rik is the editor of *;login:*.
rik@usenix.org

Four times a year, I sit down at my desk and try to come up with something different to write. I've been writing a column for *;login:* since 1996, although back then it was about Java [1], and I was not the editor. To be honest, I wouldn't still be editing *;login:* if I didn't love doing it. I enjoy working with the members and people who attend conferences, and the USENIX staff. Coming up with articles for each issue is always a challenge, one that I welcome you to help with.

Perhaps I should explain what I've been trying to do with *;login:*, since I was first asked to edit special issues on security in 1998. I actually had written a proposal to the USENIX Board of Directors in 1996 about how I would go about editing *;login:*, a proposal I found in 2006 when I was tasked with creating an updated version.

Goals

I strive to collect articles about emerging research, new tools, and techniques that I believe will benefit a broad cross-section of the USENIX membership. It's a challenging goal, as the membership ranges from industry to academic, from programmers to system administrators and SREs. USENIX has conferences on systems, security, file systems and storage, distributed systems, system administration, cloud computing, and site reliability engineering (SRE). Finding topics that may be useful to at least two of these categories, and hopefully more, has always been my goal.

Like any Program Committee member, I don't want to ever publish (or accept) bad research. In the case of *;login:*, I can often sidestep that issue because I ask the authors of accepted papers to write about their research. But that only works for topics that have been covered by papers.

When the author or authors haven't produced a paper, I look for other indications of competency in the topic at hand. I also check out references, by asking some of the many people I've gotten to know in attending USENIX conferences over the years, about the authors I have in mind.

And there's also the quality detector: you probably have one of these as well [2], and I highly recommend paying attention to any alerts it produces. For me, these alerts come when I notice that the draft I am reading fails to tell me anything useful, or covers the same territory as found online. Other times, the writer will contradict him or herself, or write things that sound just plain wrong. Again, the Web can be your friend.

Finally, timeliness is an important goal. There is lag time inherent in print publications, and I start searching for good topics six months before an issue comes out. That's a very long time when part of my goal is to cover emerging topics.

I plan on editing *;login:* into the foreseeable future and welcome new ideas and input. If you'd be interested in contributing to *;login:*'s process, and if you have suggestions for topics that should be covered, please do let us know by contacting us at login@usenix.org.

The Lineup

We start out this issue with several articles with the theme of systems. Lozi et al. had a paper at EuroSys 2016 that caught my attention, as the authors did a wonderful job of explaining how the Linux Completely Fair Scheduling system works. By adding in instrumentation and creating heat maps, they also uncovered ways in which it fails. (Those heat maps do look better in color, so you might want to view the online version of their article, or the paper it is based on.)

I ran into Tyler Harter during the 2016 USENIX Annual Technical Conference poster session. Tyler had presented a paper about OpenLambda during the HotCloud '16 workshop, and when I read the paper (I had missed his presentation), I decided that Lambdas, a way of supporting microservices, deserved a wider audience. OpenLambda is a research platform for exploring Lambdas, but Harter et al. also explain the AWS version of Lambdas and current shortcomings with this new way of providing services.

Carlos Maltzhan approached me during USENIX Security '16, wanting me to talk with some students at UC Santa Cruz about a new approach to creating papers that fosters reproducibility of CS research. He and his colleagues, Jimenez et al., explain how they have created a framework, named for Karl Popper [3], using a toolset that anyone familiar with DevOps will know about. Popper is a protocol that uses these tools, aiding in the research and paper writing process, but also creating a trail that others may follow later.

The December issue has traditionally covered security, and we do have four security-related articles.

Cittadini et al. explain another aspect of Google's BeyondCorp, a technique that does away with considering any "internal" network secure. Instead, BeyondCorp invests trust in managed devices and user authentication, and it permits access to specific services. This third article in a series on the topic explains BeyondCorp's Access Proxy, where access control lists (ACLs) control what services are available to which users using specific devices. The authors also detail the trickier aspects of making the Access Proxy work with non-HTTP protocols.

Lerner et al. had a paper at USENIX Security '16 about how Web tracking has changed over time. Their methodology involves using archive.org's Wayback Machine and examining just how accurate its record is. I was also interested in their results, but I can't say that I am surprised at just who is tracking our Web histories.

Haddadi et al. had a paper at FOCI '16 on anti-adblockers. I first thought this seemed a stretch for a workshop called Free and Open Communications on the Internet, where topics include Tor or the Great Firewall of China. If you're using adblockers, I'm sure you've encountered the manifestation of anti-adblockers; the authors explain both how these scripts work and why they are a privacy issue.

I interview Gordon Lyon, perhaps better known as Fyodor. Fyodor started working on the Nmap scanner back in the mid-'90s and has turned his passion into a successful open source business—plus a really useful tool that you should know about.

Switching over to the areas of sysadmin and SRE, we start out with an article that was adapted from a section of Allan Jude and Michael Lucas' book *FreeBSD Mastery* for inclusion in *;login:*. Jude and Lucas write about tuning ZFS for use with popular databases. While this might at first appear to be a pretty narrow topic, the authors cover the typical writing patterns used in several databases, something you may want to know if you use any of these databases, or wonder if you are using the right file system and have properly configured it.

Kurt Andersen of LinkedIn had a popular talk at SREcon16 Europe, and I asked him to reprise his talk for *;login:*. Kurt's topic is focused on how SRE teams are managed, but I think that the concepts presented would be useful in any organization that appreciates nimbleness.

Liz Fong-Jones also drew a large crowd at SREcon16 Europe with her presentation on Interrupt Reduction Projects. Working with Betsy Beyer and John Tobin, Liz created an article about how the Bigtable SRE teams worked to reduce interrupts. There are deep insights into how better to handle tickets and that may even include ignoring them while you fix underlying issues that aren't big enough to register as projects.

Dave Beazley wants to talk meta. Python has long had metaclasses, and Dave explains what metaclasses are and shows how they can be used to write cleaner Python code. Very cool ideas and mind-bending, as usual.

Dave Josephsen also goes meta, but in a different direction. Dave asks, if you rely on your monitoring, what monitors your monitoring system? Dave then describes how the people at the monitoring company he works for created a second site, one they call "dog-food," that eats the input from their main monitoring systems.

David N. Blank-Edelman wonders whether there will be weather and shows us how to use a RESTful service from Perl, not just for the current weather but for past *and* future weather, too.

Kelsey Hightower shows us how to extend Go applications with exec plugins. By creating Go programs that meet an interface, you can extend an application without editing the source.

Dan Geer considers the implications of the growth in connected devices. He projects that by 2020, there will be 6.5 IoT devices for every human alive, and that the security implications are, frankly, hard to imagine. For example, you might have heard of attack surfaces, that is, the features that make a computer vulnerable to attack. We think of attack surfaces because the goal is to reduce them. But with so many devices on the way, will this even make sense?

EDITORIAL

Musings

Robert G. Ferrell wants us to consider a new class of Web plugin: the idiocy blocker. Robert has found five main types of Internet idiots, and postulates a method for making it possible to read the comments associated with articles and other Web postings.

Mark Lamourine has written three reviews for this issue. The first two cover Single Program Applications (SPA), something I hadn't heard of before, but also something we've all encountered in the form of many online apps. Then he takes a look at a book on providing examples of practical Go programs.

Finally, Cat Allman shares her views on why she became a USENIX Board member. Cat had worked for USENIX in the noughts, and came back to help keep USENIX going. Casey Henderson has also written her yearly summary, including thank-yous to Program Chairs and the volunteers who make USENIX conferences interesting.

[1] R. Farrow, "Using Java": https://web.archive.org/web/19970606052503/http://www.usenix.org/publications/java/usingjava1.html.

[2] G. Pennycook, J. A. Cheyne, N. Barr, D. J. Koehler, J. A. Fugelsang, "On the Reception and Detection of Pseudo-Profound BS," *Judgment and Decision Making*, vol. 10, no. 6 (November 2015), pp. 549–563: http://journal.sjdm.org/15/15923a/jdm15923a.html.

[3] Wikipedia, "Karl Popper," last modified on Sept. 26, 2016: https://en.wikipedia.org/wiki/Karl_Popper.