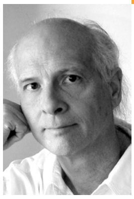
RIK FARROW

musings



Rik is the Editor of *;login:*.

rik@usenix.org

AS I WAS GOING OVER THE LINEUP OF articles for this issue of *;login:*, I found myself thinking about just how cool it is that we have not just virtualization, but services that make it easy to spin up new systems on a moment's notice. We can use the new "system" for testing, and as soon as we are done with it, it is gone. Poof.

At the same time, I found myself pondering this brave new world, facing the quandaries it creates: spinning up VMs is really easy, but how do we manage all these systems?

Greybeard

I learned system administration the way many people have—through trial and error. I had no mentors, as the people who could manage UNIX systems were still few and far between. I was fortunate in one way, though: I was being paid to learn how to manage UNIX systems and to write about it. And the people who were paying me provided systems to play with.

I fondly recall sitting in Becca Thomas's backyard in San Francisco, playing with a Xenix system while drinking a beer. My goal was to understand how dump and restor worked, with a particular focus on dump levels. The man page writer had suggested using a Tower of Hanoi sequence of dump levels, but I wanted, really needed, to know why. I couldn't just follow some unknown person's suggestions, as I knew nothing about this person's reasoning or reputation.

I wrote several system manuals for manufacturers of microcomputer-based multi-user UNIX systems, and each time I understood more. Then I ghostwrote chapters for several books before I started my own.

Now that it was my book, I had to use my own computer. I kept it locked inside a special desk that I had designed (lots of ventilation), and added deadbolts with keys for both inside and out to all house doors, to make it more difficult for someone to steal the computer. The physical security seems ridiculous to me today, as the computer really only had value to me—a thief would be hard-pressed to use a UNIX system running System V Release 3 with one megabyte of RAM and 34 megabytes of hard disk. But that was my experimental system, as well as where Thomas and I wrote. Primitive by today's standards, but a big deal back then.

A Wider World

I was missing out on a much wider world, but didn't know it. I also did consulting as a sysadmin, in those few places that needed a small multi-user system. I thought I was working in the real world of UNIX, but individual systems were quickly becoming a thing of the past.

When I proposed a title for my book, *The Handbook of System Administration*, I was stunned to discover that my publisher had contracted with another group to write a similar book, and they had already chosen that name. I later learned that this was Evi Nemeth and friends, whose fourth edition [1] of their book has just arrived, but too late for a review to be included in this issue.

Nemeth et al. were taking a very different approach to sysadmin. Their environment was the University of Colorado in Boulder, and they had access to lots of systems. From my perspective, telling people how to attach vampire taps [2] to thick Ethernet cables seemed far afield from sysadmin, yet this was an important topic in their first edition. And this pointed to something very important that my co-author and I had completely missed.

Computers would soon be connected to networks, and only rarely would they be used alone. While our book had an excellent chapter on using UUCP over dialup, they included basic IP networking. Neither book dealt with methods of managing groups of computers (beyond the files handled by Sun's NIS [3] or rdist), and for the next several years, this would remain the case. Managing multiple systems would rely on tools that could copy files from a central server to "managed" systems that were essentially all clones.

Back to the Future

Long gone are the days of having one system to manage. Before I was finished writing my first book, I was managing a development network of different vendors' workstations. My bosses did not allow me to use NIS, so just adding a user meant doing this at the console on each system.

Today, sysadmins manage tens to hundreds of systems. They obviously do not walk around to each one, login or su to root, and type commands—at least I hope not. Instead, they will use one of the many configuration management tools to handle the work for them (see the Configuration Management Summit summary in this issue, p. 104).

Jan Schaumann's article about using Amazon's EC2 as a sysadmin teaching resource is what inspired this column. Schaumann explains how important hands-on experience is in learning system administration, and his article in this issue includes links to his syllabus. Having taught sysadmin myself, I read his syllabus eagerly and liked what I saw. Schaumann makes good use of the virtual resources provided (and donated) by Amazon.

What Schaumann leaves out, for the most part, is how to manage multiple systems, perhaps with different OSes, simultaneously. I can imagine that doing so would be a topic for a more advanced class in sysadmin, as you must understand the basics before you can use tools that will duplicate your commands on possibly hundreds of systems. Actually, just thinking about setting a novice sysadmin loose with a configuration management tool is enough to make me shudder.

But learning how to manage multiple systems seems like a perfect fit for working in virtualized environments. Instead of the novice screwing up key systems, he can screw up, uh, configure, several OS instances, then learn how to clean up his mistakes. Or he can just start over, as killing off an instance and spinning up a new one erases past errors.

The Lineup

I've mentioned Jan Schaumann's article already, so let's move on to the next. Tom Limoncelli teaches us by example about satisfying customers. You might wonder how keeping water glasses topped up fits really well with different styles of handling system administration customers, but it does. Just take a few minutes and read his article.

Troy McKee takes us on an adventure where, instead of moving to the Cloud, he migrates from a hosted service. McKee covers the ins and outs of getting mailboxes and other configurations for Exchange moved from a hosted service to an internal one, with some hard-won knowledge learned along the way.

Matt Ryanczak shares some tips on finding IPv6 transit providers. Ryanczak points out that getting good IPv6 connectivity today is not unlike finding good IPv4 connectivity in 1994, as IPv6 is really a different protocol and only slowly gaining the first-class support found with IPv4. Ryanczak doesn't try to convince you to try out IPv6—he just explains some of the important steps you will need to take some day soon.

Brian Kirouac takes us down a different path, one that has become more important with the broader acceptance of smartphones. Kirouac describes how to create and use self-signed certificates to support authenticated and encrypted email for iPhones and Android-based mobiles. Chris Paget demonstrated interception of GSM voice during DefCon this summer, using homebrew equipment, leaving one to wonder if data interception can be far behind [4].

David Blank-Edelman explores places where size really does matter—those times when you need a Perl module and don't have much memory available. He takes us on a fantastic voyage with ::Tiny.

Peter Galvin suggests that we take another look at NAS. Starting with some history, Galvin contrasts NAS and SAN and provides excellent insights that may help you with your network storage decisions.

Dave Josephsen completes his series about monitoring using Argus. Josephsen demonstrates a couple of tools for extracting and sorting events or records of interest from the vast amount of flow information collected from networks.

Robert Ferrell compares airport security to network security and finds many parallels. As Ferrell writes, air travel is definitely UDP.

Elizabeth Zwicky has reviewed four books this time, starting with *Hackers*. I encouraged her to read the revised edition, and it was enlightening to read her opinions of a book I once found inspiring. Sam Stover reviews *Network Flow Analysis* and waxes enthusiastic. Brandon Ching reviewed *High Performance JavaScript*, appearing almost as excited about this book as Stover was about *Flow*.

This issue includes seven sets of reports, starting with USENIX Annual Tech and WebApps—the two main conferences from the 2010 USENIX Federated Conferences Week—followed by most of the workshops of that Week, and ending with the 2nd USENIX Workshop on Hot Topics in Parallelism.

My wife has been trying to get me to clean up my office for years now. With the advent of virtualization and services like EC2, I really don't need either of my old SPARCstations any more (plus, they are *really* slow). And the various PCs, plus extra hard drives, for running different Linuxes and BSDs seem sort of superfluous.

It is really hard to dump my SPARCstation IPC, a 25 MHz system with a 10 megabyte hard drive that cost me $6000 (with a developer's discount!) back in 1990. Perhaps I can just donate the still working RGB monitor to someone.

REFERENCES

[1] Evi Nemeth, Garth Snyder, Trent R. Hein, and Ben Whaley, *UNIX and Linux System Administration Handbook*, 4th ed. (Pearson, August 2010), 1300 pp.: http://www.amazon.com/UNIX-Linux-System-Administration-Handbook/dp/0131480057/ref=sr_1_1?s=books&ie=UTF8&qid=1280184162&sr=1-1.

[2] Vampire taps: https://secure.wikimedia.org/wikipedia/en/wiki/Vampire_tap.

[3] Naming and Directory Services (DNS, NIS, and LDAP): http://docs.sun.com/app/docs/doc/817-4843/6mkbebda4?a=view, https://secure.wikimedia.org/wikipedia/en/wiki/Network_Information_Service.

[4] "Hacker Spoofs Cell Phone Tower to Intercept Calls": http://www.wired.com/threatlevel/tag/gsm/.