

MARCUS J. RANUM

who needs an enemy when you can divide and conquer yourself?



Marcus Ranum is CSO for Tenable Network Security and a well-known security products designer and implementer. He holds the TISC “Clue” and ISSA Lifetime Achievement awards and is the author of the recently published *The Myth of Homeland Security* (reviewed by The Bookworm in this issue).

■ mjr@ranum.com

I SURVIVED THE UNIX WARS, UNLIKE most of the companies involved in them. Perhaps you remember Pyramid, SCO, Apollo, DEC, Sun, Silicon Graphics, Gould, and so on. In their day, they fought ferocious scorched-earth wars trying to win customers’ minds and money. The survivors, with the exception of Sun (a.k.a. “The Last Man Standing”) have either gone into the mists of time or are niche players forced into new markets in order to survive. Other than their conflict, what did they have in common? They were all selling some kind of UNIX operating system.

Back in the UNIX wars, the vendors had two primary axes on which they could compete: hardware speed and features of their UNIX flavor. Toward the end of the UNIX wars, a third battle evolved, surrounding the “desktop metaphor”—the look and feel of the workstations’ GUI. If you were around back then, you’ll remember the ferocious fights over whether or not the “3D-look” widgets of the Open Software Foundation (OSF) Motif metaphor were just flash and glitter or whether they were actually kind of cool. Today, few remember the argument, and the code in question would be considered remarkably tight and lightweight compared to what people now use. If you step back and look at the UNIX wars from a high altitude, the actual battlefield was very small—GUIs and features in a UNIX operating system don’t really sway customers much; the vendor who won (Sun) did so because they offered a consistent software experience (SunOS, later Solaris) across a broad spectrum of hardware at different performance levels from desktop to data center. In other words, the customers didn’t care if the GUI had a 3D look and feel as long as it was fast, reliable, and affordable.

You don’t need to be an advanced student of history to know what happened. While the UNIX vendors beat each other up over what amounted to nitpicking details, another vendor offered the same consistent kind of software experience across a broad spectrum of hardware (including laptops)—I am referring, of course, to Microsoft/Intel. Through the lens of 20/20 hindsight, it is clear that the UNIX vendors were short-sighted losers arguing over what to watch on television and fighting for the remote control while the house burned down around them.

Now, read this carefully: I am not bashing Microsoft Windows. As a UNIX system administrator with 20+ years of experience, and a Windows system

administrator since Windows 1.0, I can tell you that there isn't a whole lot of difference in the workload of efficiently running either environment. Sure, there are lots of annoying details in either one, but it takes about the same time for an expert to load and configure each system. (In the old days, UNIX machines were faster to bring online because of the prevalence of decent tape drives, while Windows was primarily loaded by floppy, but that was about the only difference.)

In other words, customers didn't "choose" Windows because it was better (or worse) than UNIX—they did it because Microsoft/Intel was careful to guarantee them a consistent software experience across a broad spectrum of hardware. And, of course, the application developers flocked to that consistent software experience because it meant their products were cheaper to develop without the headaches of version-specific differences.

In 1985, when I wrote code for my UNIX machine, it worked on all the other UNIX machines because there was basically a single flavor of UNIX, which all used the same compiler, and everything just worked. Today, you actually have to be quite careful if you want to write code that compiles and works correctly on Solaris, Linux, and BSD.

Indeed, most "open source" packages now include special tools that dynamically reconfigure the code based on complex knowledge bases that encode the differences in how Solaris says "tomato" and Linux says "tomahto." It takes longer to configure code than to compile it, these days, which is categorically not the case on Windows. Windows stuff just works and usually keeps working. Do you think that this might, just maybe, have something to do with why major apps like Adobe Photoshop, Macromedia Director, Adobe Premiere, etc., are still not available on UNIX and *never* will be?

Why is all this relevant? Because the UNIX wars didn't end and, consequently, the "last man standing" is still Microsoft/Intel.

What do I mean, "They didn't end"?

I installed Linux on one of my systems the other day so I could use it as a teaching vehicle for my class on system log analysis. But first I had to email a bunch of my friends and ask them, "What version of Linux should I use? Red Hat? Debian? Gentoo? Mandrake? Slackware? Do you think I could get away with OpenBSD or FreeBSD?" The responses I got indicated that none of my friends use the same thing but that I could be sure that if I used Flavor X some adherent of Flavor Y was going to bust my chops about it, and that someone was sure to show up with Flavor Z and have trouble making things work. Do you hear the sound of distant laughter coming from Redmond? I do.

The early days of the Linux movement were heralded with grand pronouncements of war to the death with Microsoft—war from the desktop to the data center, and a free, compatible high-performance alternative to Windows. What I see now is that the open source movement was more like a 14-year-old punk standing in the street telling Mike Tyson that he had an ass-whipping coming. Not the Mike Tyson we see today, either, but the Mike Tyson who could deliver a line-straight punch that could knock a hole through the side of a steel I-beam.

Unlike Tyson, no doubt, Microsoft was at least courteous enough to pay lip service to the threat that the 14-year-old was making, using Linux as a "credible threat" to help argue that Windows was not, in fact, a monopoly. Guys, let's face the facts: Windows is a monopoly because short-sighted open source geeks and UNIX weenies were too busy squabbling over whether RPM was better than build-from-source or Gnome versus KDE, etc., ad nauseam.

The tragedy here is that, unlike during the UNIX wars, the battlefield now is even more narrow. The hardware spectrum is a constant, so system performance is barely an issue: Nobody measures whether Slackware is faster than OpenBSD, and if someone did, nobody'd care anyhow. So the battle in the free UNIX space is entirely over command line options, system administration paradigms, installation packaging, and 3D GUI features.

I've got news for you: Real Programmers Don't Care about that garbage. Has it managed to completely escape the attention of the open source movement that Adobe, Macromedia, Corel (mostly), and so forth have blithely continued to be non-UNIX while waiting for the dust to settle? Only now they have realized that it won't settle and that oh-so-quietly the rush of announcements of support for Linux has not translated into a rush of quality applications.

Let me make a prediction for you. The open source movement is not going to hurt Microsoft to any significant degree. But it'll put Sun out of business. Good move, guys! Do I hear the sound of distant laughter from Redmond?

Is it too late to save the situation? Yes, I think it is. At this point, there are too many adherents of, and too much investment in, the "not invented here syndrome" for anybody involved in the various free UNIX flavors to come to their senses until there is only one man left standing.

But that's not going to happen because, with free software, it doesn't cost very much to remain standing forever. It's an issue of ego, not technology, so don't expect sense or sanity to kick in. We all know the expression "divide and conquer," but Microsoft didn't even need to do that—they could just sit back and watch free UNIX

fail to become a credible threat because, well, frankly, it was in the hands of egotistical detail-oriented amateurs.

Who's left that can compete with Microsoft? The place to look for alternatives is wherever there is a broad spectrum of hardware with a consistent software experience. That doesn't leave much: proprietary devices like PDAs and gaming consoles. If you want a high-impact platform that doesn't come from Redmond,

look to what the grown-ups at Sony are producing for their Playstation 2 network. The platforms are consistent and won't fragment into competing versions, because they are proprietary and the folks producing them are in business to make money, not for their personal gratification and lust for limelight. Or if you want a consistent software experience—go with Windows.

Remember: Real Programmers Don't Care.



SAVE THE DATE!
**NSDI '05, 2nd Symposium on Networked
Systems Design and Implementation**
May 2–4, 2005, Boston, MA
<http://www.usenix.org/nsdi05>

The NSDI symposium focuses on the design principles of large-scale networks and distributed systems. Join researchers from across the networking and systems community—including computer networking, distributed systems, and operating systems—in fostering cross-disciplinary approaches and addressing shared research challenges.

