We Get LETTERS

by Michael W Lucas

Hey Letters Lackey,

I keep hearing that security needs to be built in from the ground up. I'm stuck using whatever software my company says we're going to use, and a bunch of this software stinks. How can I build security from the ground up when I have to use cruddy software? I've decided the only sensible thing to do is stop caring.

—Indifferent

Dear Indifferent,

All three regular readers of this column appear to be drawn by the pleasure of watching my childish behavior when confronted with the tedious duty of writing said column. While "you insulted me in the first three words of your greeting" is a feeble justification for breaking into your systems and converting them to global-warming-accelerating SkunkCoin miners, I'm willing to make it work.

Because that's what sysadmins do. We make things work.

Even bad things.

Because software vendors insist on developing new bad things and cramming them down gullets already obscenely bloated with horrendous badness. Systems administrators stagger through the endless hours of their brief years struggling to live beneath tremendous loads of badness smelted from software like arsenic from arsenopyrite. The inherent insecurity of absolutely everything enhances this burden like a beached, deceased whale enhances an oil spill.

The urge to retreat into malaise is a natural human reaction.

Sysadmins lack the luxury of being human.

The letter writer has already surrendered, so they can stop reading now. As this column has three perverted regular readers, however, the editors insist I finish this piece with something that resembles useful advice if you don't look closely or, indeed, read it.

So:

Everything you install is your responsibility.

It might not be your fault. But it's certainly your responsibility.

You must be conversant with new software's features. When the Tyrannical Paycheck Overlord commands you to install a bucket of sewage, you must allocate time to investigate each of the floaty bits in that bucket. Making a new service merely *run* is inadequate; it must run securely. Just as you trawl through a host and exorcise unnecessary daemons, you need to sieve those daemons and disable unnecessary features. All of the BSD operating systems break up monster toolkits like PHP and Perl and even Pascal into dozens of individual packages specifically so you can choose to not install unnecessary badness. Some other operating systems install the entirety of these toolkits with a single command, giving the inexperienced intruder a banquet of badness to exploit.

With other horrible software: you don't need a feature? Turn it off. Remove unnecessary services, even inside individual packages. It's work. You'll inevitably disable features you needed and endure absurd levels of hectoring and badgering before you can re-enable them. But the work clears your conscience, and when your high-profile organization suffers the inevitable mass password snatch you'll be able to tell the charming reporter that it was entirely your boss's fault.

Developers of notably loathsome character produce software where every so-called feature is active and cannot be turned off. Programs that purport to do everything for everyone. In this worst of all possible worlds, you'll inevitably be cornered into deploying and supporting it. How does one retain the will to live despite this ineluctable destiny?

I commend system administration rules seven through ten to your attention.

#7: Temporary solutions aren't.

Whatever solution you put into place will last far, far longer than you intended or hoped. Never slap an ugly hack into place without considering its security implications. I've been employed by more than one company that had an unsecured modem for emergency access into the network. Management knew this modem existed, and specifically described eliminating it as a goal when I was hired.

I have implemented redundant VPNs and redundant bandwidth. I have integrated authentication systems never intended to interoperate. I have restructured entire networks to ensure reliable and secure emergency access through any disaster that left the datacenter running.

I have never been permitted to turn off such an unpassworded emergency modem.

The only solution is to never permit such an abomination on your network in the first place.

#8: Permanent solutions aren't.

If you stay with an organization long enough, the beautiful new solution that solves everything will gradually decay into the stinking albatross around the organization's neck.

I permanently solved my mail problems by building my own mail server and installing it at a friend's ISP. The friend moved their office. The hard drive failed from old age, so I replaced it. The friend went out of business, so I installed a virtual machine. The provider went bust.

Everything churns.

Eventually, that horrible software will churn with it.

#9: One-off solutions aren't.

Once you demonstrate that you can solve a problem, people will bring other problems to you. For solving, not for laughing derisively at. The obvious solution might be related to something you whipped up before. At one time, I maintained several dozen Perl scripts that differed primarily by the degree of stupidity in each.

Once you demonstrate that you can solve a problem, you officially own that entire class of problem. Your Tyrannical Paycheck Overlord gets to define the scope of the class.

Those one-off quick solutions need to be implemented properly, because you have to live with them.

#10: Global solutions aren't.

Those appalling all-in-one software packages are naturally alluring to financial sorts, who think that by buying them, all their problems are solved forever. These suites help solve problems, yes—primarily, the problem that the software vendor is not yet an oligarch powerful enough to demand the excruciation of all who dare question their marketing. Today these systems masquerade under names like "Enterprise Resource Planning" or "Customer Relationship Management."

And of course, they're not secure. Because why would they be, when running the management interface over telnet is so quick and easy?

You'll have no choice but to do your best to lock these systems down. And you'll wind up implementing a whole bunch of glue to make them work as best you can. The best option you have here is to place all the blame where it clearly belongs, right on the vendor of this global solution.

Don't make the mistake of caring for the vendor, though. They're in the business of selling solutions, and sysadmin rule #11 is very clear: "Solutions aren't."

Enjoy being doomed.

Michael W Lucas (https://mwl.io)'s newest books are *Sudo Mastery, 2nd Edition* and *Terrapin Sky Tango*.
If you've read this far, you might find *FreeBSD Mastery: Jails* useful.

Send your question to letters@freebsdjournal.com, and he might answer it.
If he can be bothered.