WeGet letters

by Michael W Lucas

Mister Letters Answerer,

Lots of people have built stuff on top of FreeBSD, bundled it up, and made it something you can install. I have an idea for something like that. What do you think?

—Gonna Bring My Ideas to Life

Dear GBMItL,

Interesting idea. I bet you're still young enough to remember hope.

When I built my first Unix desktop running FreeBSD 2.0 whatever I thought I would master this operating system. It had source code, and I could read C! I sincerely believed that a meticulous line-by-line assessment of the operating system's innards would grant me a skill level unknown to anyone except those anonymous faceless gurus generously volunteering their services on the freebsd-questions mailing list.

Like you, I had not yet learned to meticulously assess randomly appearing but tasty-smelling enticements for treachery.

I've written about my exploits with the FTP source code elsewhere in this column and see no need to taint your soul with a second retelling, even though extensive therapy and microdosing thrift store Shirley McClain "reveal your past life" audio cassettes have exposed further details of that odyssey. You learn that I once used a script to identify which FTP code called which function, order the source files as such, and send the result to my employer's industrial-scale printer only to discover several thousand pages later that the code was recursive would serve nicely to dispel any illusion you might suffer from that I possess any expertise whatsoever, but it would not further illuminate you on the subject of recursion or downstreams or, indeed, hope. It would suggest that experience is what you get when your employer makes you buy a new toner cartridge.

Like you, I had not yet learned to meticulously assess randomly appearing but tasty-smelling enticements for treachery.

It also did me the favor of ablating another layer off my already-stubby hope.

I've also mentioned FWTK in this very column—my fourth, if I recall correctly. Let me check my copy of available-at-sketchy-and-dubious-retail-outlets-near-you *Dear Abyss: The*

FreeBSD Journal Letters Column, Years 1-6 to verify—yes, the fourth. But I didn't talk much about it.

FWTK. The Fire Wall Tool Kit. Yes, Fire Wall. In the 1990s we had not yet invented compound nouns. Firewalls cost tens of thousands of dollars, but FWTK let you install proxies for HTTP and SMTP and more on a cheap Unix box with two network cards. All connections from inside your organization terminated on the FWTK box, which would go fetch the requested resources and hand them back to you. It supported all the modern protocols, like HTTP 1.0 and Gopher, and, through the plug interface, your ISP's Usenet server. What else did you need? It did the job reliably.

The license forbade commercial use. You couldn't resell FWTK.

But you could independently consult to install and maintain FWTK.

I was an independent consultant, wandering from company to company with my laptop and belled hat, fixing networks by application of undying principles like "use a crossover cable to connect hubs" and "PCs choked by airborne filaments fiberglass insulation will overheat and catch fire because, you know, *insulation*." The age of script kiddies had just begun and a whole bunch of local legal firms wanted one of those fancy firewall thingies. Legal firms seemed like great clients: someone who charges $250/hour doesn't flinch when a consultant charges half that. You're only claiming to be half as worthwhile as they are, after all.

Firewalls cost tens of thousands of dollars, but FWTK let you install proxies for HTTP and SMTP and more on a cheap Unix box with two network cards.

Installing FWTK was not hard. You downloaded the source code. You compiled it. You stuck the binaries in /usr/local/bin. A couple of config files later and boom—you had a firewall comparable to the big expensive ones, and best of all, the customer's money went into *your* pocket.

FreeBSD's FWTK port made the process a doddle, but was I satisfied? No, I was not. Because, you see, I had the source code. Source code makes you ambitious. Source code makes you think you can do anything. The world is stuffed with people who discovered they could read the source code and suddenly thought they could do anything, like innovate Internet payments, reinvent automobile drivetrains, build actual 1950s-style rocket ships, or reconstruct government without understanding anything about any of these. They sometimes make fortunes on the way but inevitably fail in vituperious disgrace and vitriolated shame, because being smart enough to read source code has absolutely nothing to do with competence or being a worthy human being.

I thought I could make things better. Ah, hope, such sweet toxicity!

If I could build my own FreeBSD that already included FWTK and the appropriate configuration files, perhaps even with those files in pre-initialized RCS version control, I could cut the installation time down from four hours to one. Just think of what I could do with those extra three hours! I could… search desperately for another client who needed an FWTK install, that's what I could do. Never mind that the entire point of hiring a pricey consultant is that the client gets to see them sweating for their benefit.

All I had to do was slam the source code into /usr/src/contrib, edit a Makefile and boom—when I built the operating system it would build the port. (We also didn't have freebsd-update(8) in those days. Every security patch meant building the affected components from source. FreeBSD didn't have the Open Group UNIX™ certification, but it was unquestionably chest-thumping Real Unix.

Like everything involved in the hideous nerd sport of Computer Touching, it failed.

I copied the error messages into Stack Exchange—no, wait, we didn't have such websites then. The mighty search engines of the 1990s were primarily designed for querying by fetish. The FreeBSD website had a search engine, however, and the mailing list archives were indexed. I learned, I fixed my errors, and I created new errors. Not exciting errors. Or useful errors. Just errors, to be churned through and overcome and used to create more errors.

I spent weeks of my free time on that project, where "free" was defined as "stolen from family, doing dishes, and bathing."

But once I integrated it, I would be able to run my own release, burn it to CD, and have a fully patched, installable firewall. Never mind that FreeBSD didn't yet have the release(7) man page. The mailing list archives had notes on how it was done!

I spent weeks of my free time on that project, where "free" was defined as "stolen from family, doing dishes, and bathing."

Keep in mind that I was only trying to integrate contributed software. I wasn't doing anything like, say, those HardenedBSD maniacs trying to change core kernel code while simultaneously maintaining synchronization with FreeBSD itself. On the other hand, HardenedBSD has the "advantage" of using git rather than CVS. (Making people believe that software forks are sensible, maintainable, or sustainable might be git's greatest crime, but anyway.)

I kept at it even after I ran out of law firms that needed a firewall. I wasn't going to let a stupid chunk of computer code defeat me. After all, if a guy like *the* Jordan Hubbard—who had notoriously read the wall(1) man page, thought "It can't work like that," and promptly sent a message to every single Internet user in the entire world—could do it, I surely could!

I failed.

Here's the difference between Jordan and me. He's willing, even eager, to fail at scale. In front of literally everyone.

Jordan, Rod Grimes, Nate Williams, Mike Smith, and all those folks put their work into the world and attracted other people to their vision. They were smart enough to know that the source code didn't grant them phenomenal cosmic powers but instead kept at it out of pigheadedness and the will to create something cool and useful, unlike my pigheadedness and greed.

Occasionally, they even documented that vision.

And here we are.

So, should you make a downstream? Do you have a vision? Do you have an infinite capacity for overcoming your own mistakes? Are you willing to tell the whole world what you have done? Then go volunteer for a worthy cause helping those less fortunate than you, because that's how you make changes in the world!

Yes, that involves leaving your keyboard. Sorry.

Fine. Build a downstream. See if I care. Don't repeat my mistakes, though. Go make your own. It's the best way to learn!

Also: whenever a legal firm invites you to work for them, check with other contractors they've used first. Fighting a law firm over unpaid invoices is difficult. After all, they have all the lawyers.

Have a question for Michael?
Send it to letters@freebsdjournal.org

MICHAEL W LUCAS is the author of *Absolute FreeBSD*, *Run Your Own Mail Server*, and too many other books. He probably needs his medication adjusted. Again. Learn more at https://mwl.io.