WeGetletters by Michael W Lucas

Hey Letters Column Flunky,
What's with all the firewalls? Will we ever get rid of any of them? And will you really answer any questions we send in?

Thanks,
Troublemaker

Dear Troublemaker,

Yes, I'll answer any question you ask, so long as it survives review by *FreeBSD Journal*'s esteemed editorial board. Mind you, they won't let me use words like "pusillanimous" and "mewling," so my answer might not be as useful as you might hope. They'd almost certainly reject "lily-livered," especially if I used it in reference to them, so I won't.

In fairness, I have my own rejections.

I wholeheartedly reject your question. The word "firewall" means *nothing*.

If you thawed my primordial Unix mentor from his cryogenic capsule (and handled the humdrum minutia like fixing all the cancer and starting his heart and sealing all the cells burst from ice crystals because homebrew cryogenics really translates to serious post-mortem freezer burn—especially after that three-day Great Blackout of 2003 probably drained his UPS), he wouldn't recognize anything we call a "firewall." I delved into antediluvian mailing lists to try to find the first firewall on the Internet, exercising a smidge of effort you certainly won't appreciate nearly enough, and found myself wholly blocked by this ambiguity.

A firewall started off as a type of non-flammable physical wall. Put a firewall between two buildings and you could set one to the torch without burning down the other, which must have been really convenient for the Huns when they wanted the fun of sacking Rome and setting the temples ablaze before pillaging the treasury next door. At least, that's what my predecessors told me. My 1933 Oxford English Dictionary doesn't include the word "firewall" and Oxford University knew all about Rome, so I'm guessing the early Internet engineers just made up that etymology to see if we'd believe them.

Today we've settled on a couple different approaches to firewalls: the packet filter and the proxy.

A packet filter regulates which connections can pass. You can configure a host's packet filter to protect that host or drop a packet filter in front of a whole network to control IP-level access to the network. Packet filters must be integrated with the kernel, unless you treat performance with the contempt normally reserved for politicians. FreeBSD ships with three: IPFW, IP Filter, and PF.

A proxy terminates all TCP/IP connections to the outside world, inspects the traffic at a higher level of the application stack, and originates a new request. FreeBSD includes dozens of these critters in the packages collection. A search of the ports index gives 981 proxies, and while I'm sure a bunch of those aren't actually proxies, I can't be bothered to audit the whole list, so let's go with the tediously well-known standards like Squid, SOCKS, and relayd. In a previous millennium, I made a decent living installing and supporting the FireWall Tool Kit, the primordial proxy. In my off hours, I amused myself by creating droll retronyms for FWTK.

Youngsters who use words like *devops* and *serverless* think that packet filter firewalls are the whole deal. Then their blockchain dotcom crashes. They scramble to secure insufficiently gainful employment and suffer seizures when confronted with proxies. Many globe-trampling firms disallow all unproxied connections to the Internet in the name of regulatory compliance, data control, or some nebulous hallucination of "security." Opening a direct TCP/IP connection out of one of these firms resembles splenic auto-extraction via the sinuses.

How do all of these firewally *things* get in FreeBSD?

Because someone maintains them.

Why do they maintain them?

Because they *need* them.

Nobody spends what few precious minutes our overhurried lives leave unallocated getting bludgeoned by code they don't need. I supported mod_auth_xradius for a few years because I desperately needed it to glue Apache to the company's authentication system. It was either maintain a port or run everything on the company platform, which I won't name but is alliterative with Abominable Dysentery, so I learned to send patches and deal with Bugzilla and all that, which, while occasionally frustrating, beat blue bile out of forcibly extracting useful information from Event Obscurer.

While I'm here, let me recommend FWTK. It's still online at fwtk.org. Release 2.1 came out on February 27, 1998, although a second 2.1 escaped on March 2, 1998, because we hadn't yet invented proper release versioning. FWTK is why I applauded the arrival of Squid and IPFW, which are why I celebrated IP Filter, which is why I threw a festival for the appearance of PF and relayd.

That last release is now old enough to drink and gamble in Vegas.

In related news, I'll be in Vegas on March 2, 2019. Perhaps I should throw FWTK a coming-out party.

> **Dear Letters Column Flunky,**
> **I meant the packet filters, you silly goose. And, if you'll answer any question: What's the difference between a poorly-dressed programmer on a unicycle and a well-dressed programmer on a bicycle?**
>
> **Troublemaker**

Dear Troublemaker,

Again: it's because people need them.

Any code in FreeBSD, kernel or userland, needs care and feeding. Programmers get these daft ideas like "support new hardware" and "nobody uses twoax any more," so they constantly change code internals and APIs in the name of progress. Change the network stack to support 40GB Ethernet cards and someone has to assess the packet filter code to see if it still works.

If nobody tweaks that code, eventually it no longer works and someone—traditionally, a Dane—axes it from the tree.

IPFW is the primordial FreeBSD firewall. It's a favorite among many senior developers who learned it in the late nineties and don't see why anyone would want anything simpler. I've used it to simulate a transoceanic link in a local office, because web developers should suffer the same fate as their hapless users.

IP Filter is for those condemned souls who must use a single packet filter on multiple flavors of Unix. I don't know what they did to be sentenced to multiplatform torment, but it must have been appalling even by my exquisitely high standards.

PF is by far the most popular general-purpose packet filter. It was ported from OpenBSD and then forked to handle FreeBSD's kernel locking, so don't trouble yourself to ask the maintainers when a new import from OpenBSD will happen. It won't. My repeated but wholly unscientific surveys show that roughly 80% of FreeBSD users who run packet filters use PF.

PS: Attire.

Michael W Lucas (https://mwl.io) is the author of too many books, including the brand new third edition of *Absolute FreeBSD*, *PAM Mastery*, and *Butterfly Stomp Waltz*. George Neville-Neil bribed him to write this column and Lucas is still awaiting payment. Send your questions to letters@freebsdjournal.com. Letters will be answered in the order in which they befuddle, betray, or bewilder the columnist, and might be edited for his own beguilement. •