



27 Years with the Perfect OS

FreeBSD is perfect and I have been using it for a bit over 27 years. But FreeBSD is not the only OS on my desktop.

BY PETER CZANIK

If you are a longtime FreeBSD user, you probably know everything I have to say, and, what's more, you can probably add a few more points. But hopefully, there will be some Linux or even Windows users among readers who might learn something new!

FreeBSD is not just a kernel but a complete operating system. It has everything to boot and use the system: networking utilities, text editors, development tools and more. Why is that a big deal? Well, because all these components are developed together, they work perfectly together! And a well-polished system is also easier to document. One of my favorite pieces of documentation is the [FreeBSD Handbook](#) which covers most of the operating system and is (most of the time) up to date.

Of course, not everything can be integrated into the base operating system, and this is where FreeBSD ports and packages can be useful. The ports system allows a clean separation of the base system and third-party software which allows you to install third-party software on top of a FreeBSD base system.

There are tens of thousands ready-to-use software packages to choose from. For example, all the graphical desktop applications are in ports, just as various web servers or more up-to-date development tools.

FreeBSD is flexible. It runs on anything from Raspberry Pi through desktop machines to high-end servers. You can use binaries provided by the FreeBSD project for both the base system and packages. But you can also recompile everything and carefully customize to your own environment. It's really no wonder so many appliances are FreeBSD-based.

The engineering of FreeBSD is fantastic. All small aspects of the operating system are carefully designed before implementation which results in perfect solutions in most cases, but also means slightly slower progress. If you like to use the latest and greatest hardware as your desktop, it might not yet be fully supported—if at all. This is why many people consider FreeBSD a server OS (including me), even if FreeBSD runs perfectly on the desktop—on older hardware.

FreeBSD is flexible. It runs on anything from Raspberry Pi through desktop machines to high-end servers.



The University Years

When I started university 27 years ago, the facility already had a FreeBSD server—a 486 box with 16 MB (not GB) of RAM and an SCSI hard drive. I do not recall the exact version of FreeBSD, but it was still 1.X, as version 2.0 was released only months later. It took many days to download the new version: our whole university had a 64K line at that time.

There was no Linux at the facility, and it became my task to install the first Linux server, which gave me the opportunity to see the early days of both operating systems next to each other. When it came to number of installed systems, Linux quickly won with its “good enough” attitude, as the always perfect FreeBSD was often slower to adopt new hardware or technologies. However, when it came to work, the well-designed, no-surprise implementation of FreeBSD was and still is a lot more pleasant experience—at least for me.

In the first two years, I was a regular FreeBSD user, but for another sixteen years, I also maintained the server, even after I had left the university. FreeBSD was famous for its stability, and even long after Gmail became widely available, many students and faculty asked for a username on that server.

Being Jailed!

Fortunately, it was not me, but the web servers! I had a part-time sysadmin job and was running web servers. Serving static pages is not scary but serving PHP pages takes some courage. Luckily, just when I needed to solve PHP serving for customers, jails were introduced to FreeBSD.

At first, I had a single server, and all the jails were created and configured by hand. That is not a huge problem when you can count your customers on one hand. But it becomes quite problematic when you have multiple servers and dozens of customers. So, I introduced a couple of shell scripts, later we introduced central management, LDAP, and a Windows-based management app and almost everything could be automated.

While many hosting companies around us continuously reported breaches affecting multiple customers, using a well-hardened, FreeBSD base system and self-build, and hardened jails on top did the trick for us. Of course, even the best hardened jail environment cannot help on a badly configured WordPress instance. Quite a few web servers were defaced but this consistently only impacted a single jail. That’s not bad when you have hundreds of jails running on a single server, and at the peak, there were dozens of physical and virtual machines in the cluster. Everything was compiled by me on these servers, and I removed all options from the base system that were not mandatory for running the jails. Software inside the jail was hardened both at compile time and by configuration.

Once I left the company, the same system stayed in use for another five years without any updates. They carefully monitored system logs, and before shutting down the whole system, I got access one more time for an audit. After spending a couple of hours checking the last remaining hosts, I could not find any evidence of a security incidence. FreeBSD jails are fantastic!

After spending a couple of hours checking the last remaining hosts, I could not find any evidence of a security incidence.

FreeBSD jails are fantastic!



The syslog-ng Years

When I joined my current workplace, one of the first tasks was to make sure that Linux distributions and FreeBSD had up-to-date syslog-ng packages. Getting a package updated is much faster if, in addition to asking for it, I also provide an updated package to the package maintainer. So, I learned the basics of FreeBSD ports from the maintainer point of view.

I am not a FreeBSD ports committer, as I only work on a single package, but I work closely with a committer and this arrangement is easier for both of us: I know the syslog-ng part better, so I can change the port to enable new features. He knows FreeBSD ports a lot better than I do and can make sure that the syslog-ng port conforms to the latest recommendations about ports.

Ten years ago, at FOSDEM, I spent part of my time at the BSD devroom. There was a talk about how to extend one of the FreeBSD-based appliances with additional packages. After the talk, I asked how syslog-ng could be integrated. I even gave my business card to the speaker. I was never contacted, but soon after that discussion, I discovered that FreeBSD-based appliances started to feature syslog-ng for logging.

Syslog-ng was known for its portability. Over the years, all the supported commercial UNIX variants disappeared and the developer team focused on Linux. My regular testing on FreeBSD helped to ensure that syslog-ng did not turn into a Linux-only software.

A couple years ago, I learned about BastilleBSD, a jail management system for FreeBSD. Remembering the pain of implementing my own scripts two decades earlier, I really appreciated the features and ease of use that BastilleBSD provided. It now has a template system—similar to Dockerfile in the Linux world—to make creating jails easier. There is also a template for syslog-ng. You can read more about it at: <https://www.syslog-ng.com/community/b/blog/posts/running-syslog-ng-in-bastille-revisited>

What's Next?

Occasionally, I give a try to FreeBSD on the desktop, but then I give up quickly. I love state-of-the-art hardware but unfortunately FreeBSD does not. As an example, Windows and Linux run without problems on my AMD Ryzen 5800 + nVidia 3070 system while FreeBSD runs only in text mode—and I could not get graphics to work. So, for me, FreeBSD remains a server operating system and I really love it. And, once I have some real servers again—and not just virtual machines for development and testing—I look forward to running FreeBSD on them!

PETER CZANIK started using FreeBSD with version 1.X in 1994. He is an engineer working as open source evangelist at Balabit (a One Identity business), the company that developed syslog-ng. He assists FreeBSD and Linux distributions to maintain the syslog-ng package, follows bug trackers, helps users, and regularly speaks about sudo and syslog-ng at conferences (SCALE, All Things Open, FOSDEM, LOADays, and others). In his limited free time, he is interested in non-x86 architectures and works on one of his PPC or ARM machines.

Occasionally, I give a try
to FreeBSD on the desktop,
but then I give up quickly.