By Michael W **LUCAS**

The Daemon's Dozen

Introducing FreeBSD 12.0

We've all been eagerly awaiting FreeBSD 12.0 since about two hours after FreeBSD 11.0 escaped. Here I'll take a few looks at some new features, removed features, and forthcoming changes, starting with storage.

ZFS users are probably familiar with the beadm(8) package, the boot environment administration program. We've previously covered it in the *FreeBSD Journal*. FreeBSD now includes bectl(8), a boot environment control program. It retains much of the beadm(8) syntax, letting you create, activate, list, and destroy boot environments. Additionally, it lets you mount unused boot environments and create jails from those boot environments. I've used jails to troubleshoot problematic boot environments, so this is a welcome addition to the base system.

Many people have used GELI to encrypt disks. Doing hard drive encryption properly is tricky, but GELI handily reduces the risk of exposing confidential data when your laptop gets stolen. GELI was painful to use on systems with many hard drives, though. Each disk needed to be configured separately. With FreeBSD 12.0, GELI can now configure multiple disks simultaneously so long as they use the same encryption key, easing this annoyance for sysadmins who administer such machines. Also, in the GELI realm, the installer can now encrypt disks on hosts booting off of UEFI.

FreeBSD's GEOM system lets us stack storage transformations on top of one another in exactly the way you need for your application, hardware, and environment. If you want a single massive striped and encrypted storage array, you could stripe the disks together and then encrypt them—or you could encrypt each disk separately and then stripe them. GEOM empowers you. The catch, of course, comes when you're trying to figure out what you did with that power. The new geom `-t` command displays the host's GEOM stack.

geom -t

```
Geom                    Class        Provider
ada0                    DISK         ada0
ada0                    DEV
ada0                    PART         ada0p1
  ada0p1                DEV
  ada0p1                LABEL        gpt/gptboot0
    gpt/gptboot0        DEV
ada0                    PART         ada0p2
  ada0p2                DEV
  swap                  SWAP
ada0                    PART         ada0p3
  ada0p3                DEV
  zfs::vdev             ZFS::VDEV
```

You can easily see how the providers and consumers get stacked onto each other, what goes where, and the labels applied to everything.

Similarly, if you want to see all of the information about a particular GEOM container, use the new geom -p command. Here I tell GEOM to spill everything it knows about da2p1.

geom -p da2p1

Some storage changes are not so obvious. A TRIM request is used to tell an SSD or a memory filesystem that a block is no longer used. FreeBSD 12.0 batches and consolidates those requests, reducing disk I/O. You'll see dozens of such minor performance enhancements throughout the release.

FreeBSD 12.0 has one major new feature for storage, though: a parallel NFS (pNFS) server. Parallel NFS can greatly enhance NFS performance but requires a pNFS-aware client. FreeBSD's NFS client is pNFS-aware. It's a brand-new feature, though, and rather than telling you to deploy it immediately I'd encourage you to thoroughly test it in your environment. pNFS also requires NFSv4, so if you're still using an older version of NFS it's time to look at upgrading.

NFS touches on the network, so let's talk networking next.

The FreeBSD 12.0 kernel enables the virtual networking stack VIMAGE by default, allowing you to use multiple routing tables. You can now have your host use its standard routing table but create a new routing table just for jails running on that host. Also, you can now use PF within a jail, and PF's HFSC traffic shaping can now handle up to 100 Gbs.

Eight-bit 10Mbs network cards are no longer popular enough for FreeBSD to support, and their presence in the kernel creates roadblocks for development of current features. Drivers for purely 10Mbs cards like lmc, ixgb, and nxge have all been dropped from the current kernel. Many drivers for 10/100 Mbs cards such as ed, ep, pcn, and so on are deprecated and salted for removal in FreeBSD 13.0. Twenty years ago, I was a big fan of NE2000-compatible cards. They were essentially indestructible. The problem is that was twenty years ago. It's time to buy a new network card.

FreeBSD has had a long-running problem with video drivers. Video cards come out more frequently than FreeBSD releases. Users sensibly want to use all the fancy features of their new video cards and would get frustrated when a two-month-old FreeBSD didn't support those drivers. FreeBSD solved this by removing the advanced video drivers from the base operating system and making them a port instead. Ports are much more easily updated than base system components, meaning users get accelerated video and 3D support and all the other fancy stuff they want. Install the drm-kmod package for the latest video drivers and see the FreeBSD Graphics Wiki at https://wiki.freebsd.org/Graphics for details on how to use it.

If you need to extract information from last(1) and lastlogin(8), these programs now support libxo. Try last --libxo xml to get a sample of the output and use your organization's standard XML tools to gather the information you need.

There's a bunch of other new features and details. The boot loader now includes Lua rather than Forth, so all you Lua hackers can customize the boot process. Some older, standard device drivers have undergone improvements. While the console could previously display text faster than I can read, the vt(4) driver now runs up to four times as fast as in FreeBSD 11.0. Contributed programs like ntpd(8) and mandoc(1) have been upgraded. The random number generator random(4) has seen modernization, and the integrated OpenSSL will get support for several years. The default pager is now less(1), and newsyslogd(8) ignores setuid and executable log files—as it should.

FreeBSD 12.0 has many smaller enhancements, improvements, and tweaks. Really, the best way to see them all is to install it and see what happens. I've been happy with it. I think you will be as well.

Michael W Lucas has been using BSD since the 1980s. His most recent book is the third edition of *Absolute FreeBSD*. Learn more at https://mwl.io.