

FreeBSD 15.0: Fixes and Features

BY JOHN BALDWIN

The FreeBSD community continues to push forward with the release of 15.0. This release includes numerous features, refinements, and bugfixes relative to 14.0, which was released in November of 2023. Highlights of some of the changes are listed below, but more details can be found in the [release notes](#).

Improving Project Structure

While FreeBSD's developers have merged many patches over the last two years, they have also refactored several of the project's processes and structures. These changes aim to streamline development workflows and make optimal use of developers' time.

Colin Percival proposed several changes to FreeBSD's release schedule shortly after the release of 14.0. As he detailed in the journal [earlier this year](#), the new schedule features a fixed cadence of both major and minor releases. 15.0 is the first major release following the new schedule.

At last year's BSDCan, FreeBSD's core team announced the new srcmgr team to manage the source repository. Delegating tasks such as src commit bits to this new team allows the core team to focus its efforts on strategic planning for the project as a whole.

15.0
This release includes numerous features, refinements, and bugfixes relative to 14.0.

Packaged Base System

The [pkg\(8\)](#) tool has proven itself as a mature system for managing binary packages. FreeBSD has been using pkg(8) to manage third-party packages built from the ports collection for many years. Over the past few years, a group of developers has worked to provide binary updates to the base system using pkg(8). This has included enhancements to the pkg(8) tool as well as changes to the base system build process to integrate with the pkg(8) tool. 15.0 will be the first major release supporting binary updates via pkg(8). The older system of distribution sets managed by freebsd-update(8) will also be supported in 15.x, allowing end users a graceful transition between these systems. FreeBSD's developers expect to switch the installer to use a packaged base system in the next major release.

Focusing Development Effort on Future Systems

Developer time and effort are scarce resources. To provide a high-quality system, FreeBSD has long focused on contemporary, widely deployed systems. Over the past major releases, FreeBSD has chosen to deprecate support for older CPU architectures that are seeing declining use in industry and limited developer support. 14.0 deprecated several 32-bit architectures that will not be supported as a standalone architecture in 15.0, such

as 32-bit x86 and 32-bit PowerPC. The 64-bit versions of both architectures will continue to support running 32-bit binaries in 15.0 and beyond. However, 32-bit kernels for these architectures are no longer supported in 15.0, and release artifacts such as install images will not be provided for 15.0.

Networking

15.0 includes support for new networking devices as well as improvements to TCP. Nvidia contributed changes to support inline IPsec offload, enabling smart NICs to offload IPsec encryption/decryption from the host CPUs to the NIC. This is similar to kernel TLS offload, but for IPsec. The mlx5en(4) driver supports IPsec offload on ConnectX-6 and later adapters. Local (UNIX domain) sockets were refactored in 15.0, resulting in increased throughput and reduced latency for local stream sockets.

Storage

Several new storage features are included in the upcoming release. Samsung contributed a driver for the Universal Flash Storage standard, an alternative to the eMMC standard used for embedded flash storage. The driver's author, Jaeyoon Choi, covers this in more detail in "Universal Flash Storage on FreeBSD" in this issue. 15.0 also includes support for NVMe over Fabrics using the TCP transport as covered in a previous [journal article](#). Since that article was published, support for NVMe-oF has been merged into the [ctld\(8\)](#) daemon, and the nvmfd daemon has been removed.

Also included in 15.0 is a native implementation of the [inotify\(2\)](#) family of system calls. This implementation is API-compatible with the same system calls as Linux and is available for both native FreeBSD binaries and Linux binaries running under the [Linux compatibility layer](#). For many use cases, inotify(2) is both more reliable and more efficient than EVFILT_VNODE kernel events available via [kevent\(2\)](#). It is also a widely used API in existing desktop software such as KDE.

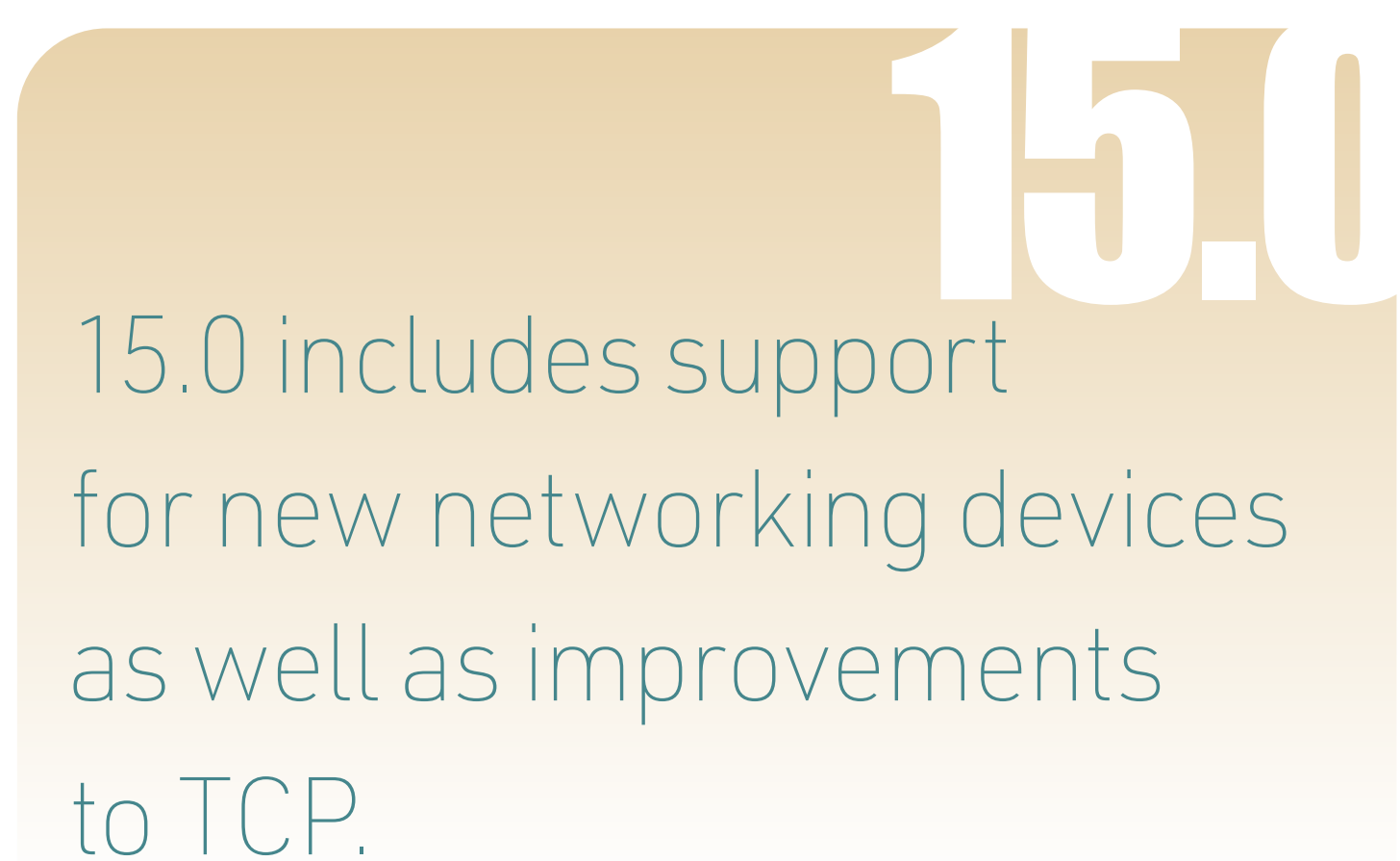
Virtualization

FreeBSD's type 2 hypervisor, bhyve, includes several updates in 15.0 as well. Both the in-kernel monitor and the userspace hypervisor are now supported for the 64-bit ARM and RISC-V architectures. A few advanced features, such as PCI pass-through, are not yet supported, but both FreeBSD and Linux guests using existing bhyve device models, such as VirtIO, are fully supported on both new architectures.

In addition to increased architecture support, bhyve can now use the net/libslirp package to provide a userspace backend for network devices. This allows the host to connect to guests over a network connection without requiring additional host network configuration, such as [tap\(4\)](#) devices.

Architecture-Specific

15.0 includes a processor tracing framework, [hwt\(4\)](#), which collects streams of events logged by CPUs. These events include details about software execution, such as control



flow changes, exceptions, and timing information. The framework supports events logged by ARM's Coresight and Statistical Profiling Extension (SPE) and Intel's Processor Trace (PT).

This release also includes support for AMD's IOMMU, which is particularly useful on systems with many cores. IOMMUs on x86 systems provide several features. The main purpose of an IOMMU is to provide an alternate address space for device DMA requests, which is useful both for virtualization (such as PCI pass-through) and for security (restricting memory access for untrusted devices). On x86, IOMMUs also interpose on interrupt delivery, permitting device interrupts to be routed to CPUs with numerically larger IDs. Previous releases of FreeBSD have included support for Intel's IOMMU (DMAR), and 15.0 introduces support for AMD's IOMMU.

Extended Error Reporting

Traditionally, in POSIX systems, system calls report errors during execution by returning an integer error code. This error code is available in the special global variable `errno` and can be translated to strings in functions such as [strerror\(3\)](#). 15.0 introduces a new extended error facility in the kernel, which saves additional information about an error, including an additional string description and the location in the source code of the error. The string description can be retrieved after a failed system call via the `uexterr_gettext(3)` function. The [err\(3\)](#), [errx\(3\)](#), [warn\(3\)](#), and [warnx\(3\)](#) family of functions will include the extended string description in the messages output to `stderr` automatically. Extended error information is also available via [ktrace\(1\)](#).

Third-Party Software

FreeBSD's base system includes several components that are externally maintained. As with every release, 15.0 updates many of these components by importing newer versions of the upstream software. The list of updates is too long to mention here, but a few deserve special mention. OpenZFS has been updated to the latest release, 2.4.0. OpenSSL has been upgraded to the current long-term support release (3.5), ensuring upstream support for the life of the stable/15 branch. The current version of MIT Kerberos has been imported into the base system, replacing the older Heimdal implementation. Toolchain utility programs such as [ar\(1\)](#) and [size\(1\)](#) are now provided by LLVM rather than the ELF toolchain project. This enables support for link-time optimization (LTO) in the base system toolchain.

Conclusion

FreeBSD 15.0 incorporates fixes and features contributed by a broad community over the last two years. Thank you to everyone who has contributed to this release by testing snapshots, reporting bugs, submitting patches, working with users on social media, and performing countless other tasks. We hope you enjoy FreeBSD 15.0. Please join us as we continue to move forward with FreeBSD 16 development!

JOHN BALDWIN is a systems software developer. He has directly committed changes to the FreeBSD operating system for over twenty years across various parts of the kernel (including x86 platform support, SMP, various device drivers, and the virtual memory subsystem) and userspace programs. In addition to writing code, John has served on the FreeBSD core and release engineering teams. He has also contributed to the GDB debugger. John lives in Concord, California with his wife, Kimberly, and three children: Janelle, Evan, and Bella.