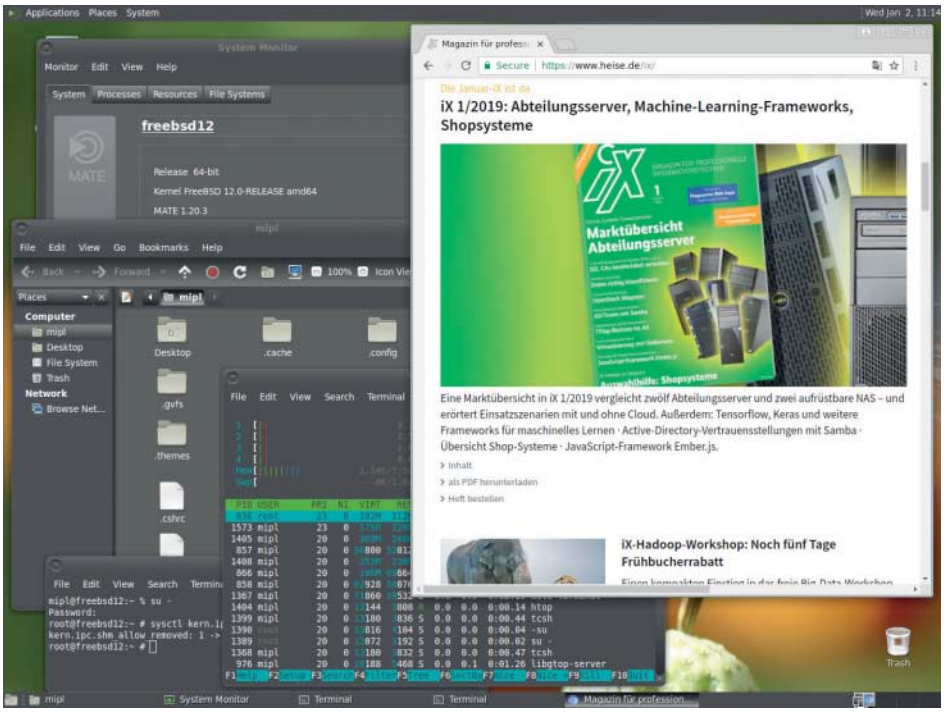


FreeBSD 12.0 erlaubt VMs in Jails

Umzäunt

Michael Plura

In FreeBSD 12 verbesserten die Entwickler viele Bereiche, darunter Sicherheit, Treiber und Virtualisierung.

Mit einer Optimierung für Systeme mit AMDs Zen-Architektur, also AMD Epyc-Server und Ryzen-Desktops, präsentieren die FreeBSD-Entwickler Version 12 ihres UNIX-Derivats. Eine Installation auf amd64-Hardware erfolgt standardmäßig mit aktivierter NUMA-Unterstützung (Non-uniform Memory Access), was Zugriffe auf den direkt an die jeweilige CPU angebundenen Speicher priorisiert. Der Scheduler kann Prozesse auch auf asymmetrischen NUMA-Architekturen verwalten.

FreeBSD 12 nutzt auf Wunsch AMDs Hardwareverschlüsselung, die soll jedoch im Vergleich zu Intel AES-NI leistungsschwächer sein. Wichtig für Spectre/Meltdown: Microcode-Updates für Intel-CPUs pflegt das System vor dem Laden des Kernels in die CPU ein. Aktuelle Grafikkartentreiber für moderne ATI/AMD- und Intel-Chips hält das Projekt nun unter anderem in der Ports-Collection vor.

Bhyve-VMs im Jail

FreeBSDs nativer Hypervisor Bhyve erhielt Optimierungen für AMD SVM (Hardwarevirtualisierung), und CPU-Topologien kann man nun aus dem Userland heraus definieren. Bei aktiviertem *sysctl security.jail.vmm_allowed* lassen sich Bhyve-VMs in Jails (Container) einsperren. Dieser Schutz verhindert Schäden am Hostsystem, sollte ein Angreifer aus einer virtuellen Maschine ausbrechen.

Innerhalb der Jails lässt sich endlich der Paketfilter *pf* für Firewall-Appliances nutzen. Auch FUSE-Anbindungen (Filesystem in Userspace) sind nun möglich. Neu ist die *virtio_scsi(4)*-Unterstützung sowie eine NVMe-Emulation. Weitere Anpassungen verbessern die Funktion von FreeBSD-12-Instanzen in virtuellen Umgebungen wie Amazons EC2.

Multiple TCP-Stacks

In FreeBSD 12 lassen sich mehrere TCP-Stacks definieren und separat voneinander konfigurieren. Statt den gesamten Datenverkehr durch einen TCP-Stack mit verschiedenen Netzwerkadaptern zu schleusen, kann man so für die jeweilige Aufgabe optimierte TCP-Stacks einrichten. Laufen beispielsweise SAN- und iSCSI-Stacks getrennt von Heartbeat-Stacks für CARP oder einzelnen Subnetz-/VLAN-Segmenten, dürfte das nicht nur der Geschwindigkeit, sondern auch der Sicherheit zugutekommen. Ein solches Feature bieten sonst nur Systeme wie QNX, IBMs zOS oder VMware vSphere 6.0.

Neu ist ein Treiber für Microchips USB-3-GBit-Ethernet-Adapter LAN78xx, dafür entfallen etliche 10/100-MBit-Treiber, etwa für alte NE-2000- und 3Com-Karten. Auch vom Token-Ring- und dem ARCNET-Protokoll hat sich das Projekt getrennt. TRIM ist nun für alle mit UFS/FFS und ZFS formatierten SSDs eingeschaltet.

FreeBSD, (k)ein Nischenprodukt

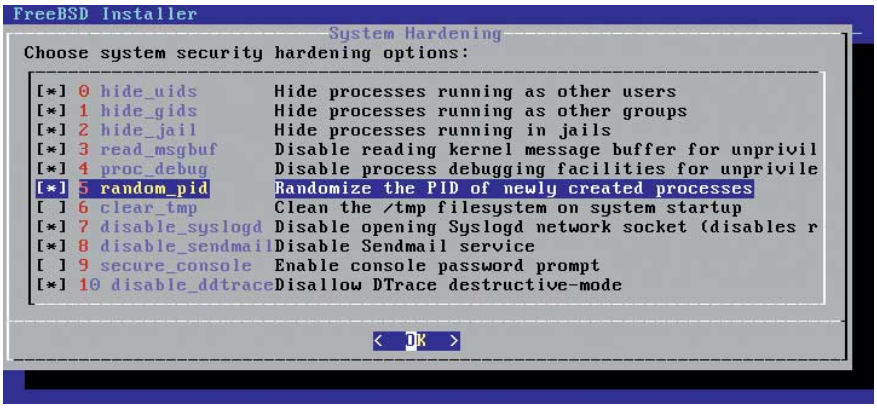
Unter anderem Unternehmen wie Apple (große Teile von macOS, iOS), Juniper (JunOS), Sony (PS3/PS4), iXsystems (TrueNAS/FreeNAS) und QNAP (QES) setzen FreeBSD ein. Die Konzentration auf einen stabilen und sicheren Netzwerkbetrieb sowie eine konservative Entwicklung bringen klare Vorteile für Serverdienste und Appliances – neben der BSD-Lizenz ein Grund für Internetdienstleister wie Netflix oder WhatsApp, sich für FreeBSD zu entscheiden. Natürlich ist FreeBSD 12 auch als Desktop einsetzbar, jedoch im Vergleich zu anderen Systemen mit deutlich mehr Konfigurationsaufwand.

Im Gegensatz zu GNU/Linux existiert keine Trennung zwischen Kernel und Distribution, bei FreeBSD sind Kernel- und Userland aus einem Guss. Die Installation endet daher bei einem Basissystem, weitere Dienste oder Desktops sind manuell zu installieren und zu konfigurieren. Das geschieht bei allen BSDs entweder über fertige Binärpakete oder durch Kompilieren von „Ports“, was Anwendungen speziell auf die eigene Systemkonfiguration und Hardware optimiert. Es gibt auch einige vorkonfigurierte FreeBSD-Systeme wie TrueOS/GhostBSD (Desktop), pfSense/OPNsense (Router/Firewall) und FreeNAS/TrueNAS (NAS).

Die Entwicklung von FreeBSD 12 erfolgt komplett mit Clang/LLVM, LLD, LLDB, compiler-rt und libc++. Die Entwickler aktualisierten die Umgebung komplett auf Version 6.0.1 und entfernten als letztes GNU-Werkzeug den GNU Debugger GDB aus dem Basissystem. Reproducible Builds erlauben jederzeit reproduzierbaren Binärcode.

FreeBSD ist die Referenzplattform für das leichtgewichtige Sandbox-Framework Capsicum, mit dem sich Zugriffe auf Namespaces einschränken lassen. Capsicum ist jetzt auf weiteren Architekturen aktiv und in viele Serverdienste wie *sshd(4)* eingepflegt. Neu ist eine Manualpage zu *arch(7)*, die Unterschiede der ABIs (Application Binary Interface) aller verfügbaren CPU-Architekturen und Plattformen dokumentiert.

Die Installation läuft im Textmodus, ist leicht durchzuführen und kann auch verschlüsselt auf ein ZFS-Volume erfolgen. Etliche Sicherheitsfunktionen lassen sich am Ende der Installation aktivieren (siehe Screenshot). Updates erfolgen per *freebsd-update fetch/install*, die Installation von Software binär über *pkg install* … oder per *make* aus den Sourcen.



Gehärtet: Wichtige Sicherheitsfunktionen des FreeBSD-12-Kernels lassen sich bei Bedarf am Ende der Installation einschalten.

Dienste aktiviert und konfiguriert der Administrator über Einträge in */etc/rc.conf*. Manualpages und mitgelieferte Dokumentation sind komplett und werden akribisch gepflegt, sodass FreeBSD mit Bordmitteln administrierbar ist.

Das weitgehend POSIX-konforme FreeBSD-12.0-RELEASE steht unter der freien BSD-Lizenz und ist als ISO-, USB-Stick- und SD-Card-Image für die Architekturen amd64, i386, powerpc, powerpc64, powerpcspe, sparc64, armv6, armv7 und aarch64 herunterladbar. Cloud-Images gibt es auf Amazon EC2, Google Computer Engine und für HashiCorp/Atlas Vagrant. (avr@ix.de)

Michael Plura

lebt in Schweden und ist freier Autor mit den Schwerpunkten IT-Sicherheit, Virtualisierung und freie Betriebssysteme.