Today's FreeBSD

Michael W Lucas

https://mwl.io

For mug.org 9 October 2018

About Me

- Author
- Unix since 198(mumble), sysadmin since 1995
- Founding member of South East Michigan BSD User Group, semibug.org
- Blatant BSD demagogue
- Author of many tech books, including *SSH Mastery*, *Cisco Routers for the Desperate*, and the brand-new 3rd edition of *FreeBSD Mastery*
- As Michael Warren Lucas, novels like *git commit murder*
- Martial arts, pet rats, married, blah blah blah

Competition is the Best

- “Competition” has become a dirty word
- I’m not running down your favorite OS
- I will mention places where FreeBSD beats out competitors
- Your favorite beats out FreeBSD somewhere as well
- Devs hang out together at cons
- Each BSD competes with the others, BSD competes with Linux, open source Unixes compete with commercial Unixes…

What is FreeBSD?

- A freely available Unix-like operating system
- Derived from original AT&T Unix
- Founded 1992
- Primary platforms: amd64, i386
- Up-and-coming: arm64
- Moves 25% of the Internet’s traffic

BSD Licensing

- Don’t claim you wrote this
- Don’t blame us if it breaks
- Don’t use our name to promote your product

- Yes, it’s commerce friendly. So?

Who uses FreeBSD?

- ISPs
- Thousands of small orgs
- Netflix
- Dell/EMC
- Juniper
- NetApp
- Sony
- Apple

Who Builds FreeBSD

- About four hundred committers
- Thousands of contributors
- Millions of users

FreeBSD Features

- Complete operating system
- Fast networking, SCTP, IPSEC, IPv6, etc
- Many filesystems
  - #INCLUDE_LUCAS_MUG_ZFS_20180410
- GEOM and encryption
- Integrated package management and building
- Binary updates & upgrades
- Runs Linux binaries
- Lightweight virtualization
- Documentation

FreeBSD versions

- Currently have 10.4-RELEASE and 11.2-RELEASE
- 12.0 is coming soon

- That sounds simple, but you'll keep seeing –STABLE and –CURRENT

FreeBSD-current

- The tender edge, the latest code
- Might explode. Probably not.
- Might destroy your data. Probably not.
- Install images rolled every few weeks
- A straight line of development, occasionally renumbered
- Today, 12.0-CURRENT is being polished for release
- Soon, 12.0-RELEASE will fork off –current.
- -current will be relabeled 13.0-CURRENT

FreeBSD-stable

- Changes that don’t churn the system get backported from –current to recent release.
  - Don’t change abi/kbi
  - Don’t violate POLA
- 12.0-stable = 12.0 plus backports
- Eventually, 12.0-stable becomes 12.1-RELEASE

Releases

- 12.0-RELEASE will get security and stability fixes
- Identified by patchlevel, such as 12.0-RELEASE-p9

System Configuration

- All core functions enabled and disabled in /etc/rc.conf or /etc/rc.conf.d/

```
hostname="mail.michaelwlucas.com"
ifconfig_vtnet0="104.236.197.233 netmask 0xffffc000"
defaultrouter="104.236.192.1"

sshd_enable="YES"
ntpd_enable=YES

clear_tmp_enable=YES

named_enable=YES
named_flags="-t /var/named"
```

- Can edit by hand or sysrc(8)

Networking

- BSD had the first TCP/IP implementation
- Designed to accommodate new network stacks
- Stream Control Transport Protocol early implementation
- Early IPv6 implementation through KAME
- Aggregation & VLANs, of course
- Can dynamically change TCP congestion control algorithms
- Does drop old stuff

Storage with GEOM

- Sometimes a mirror of RAID-1 makes sense: sometimes striping across multiple mirrors; sometimes, mirroring RAID-5.
- Do you encrypt on the drive, or encrypt the RAID container?
- What is the physical arrangement of your storage hardware?
- GEOM's storage rule: you know what you're doing
- Each GEOM layer is a device node. Install a crypto geom on a disk, then put RAID on the disk's crypto device? Or put the RAID on the disk and slather crypto on the RAID disk?

Storage Encryption

- GELI and GBDE

- GELI – friendly happy encryption for your laptop and financial servers
  - Multi passphrase support for corporates
  - Key files on USB drives
- GBDE – protects the user from threats
  - Hide data in other filesystems or swap space
  - “The passphrase is correct, but the data has been destroyed”

Base system vs Packages

- FreeBSD ships as a complete operating system, not packages
- Distinct separation between packages and OS
- /usr above /usr/local is pretty much for the OS
- /usr/local is for packages
- /home is yours
- / somewhat replicated under /usr/local : /usr/share/examples -> /usr/local/share/examples, /etc -> /usr/local/etc, and so on

What Goes Into Packages?

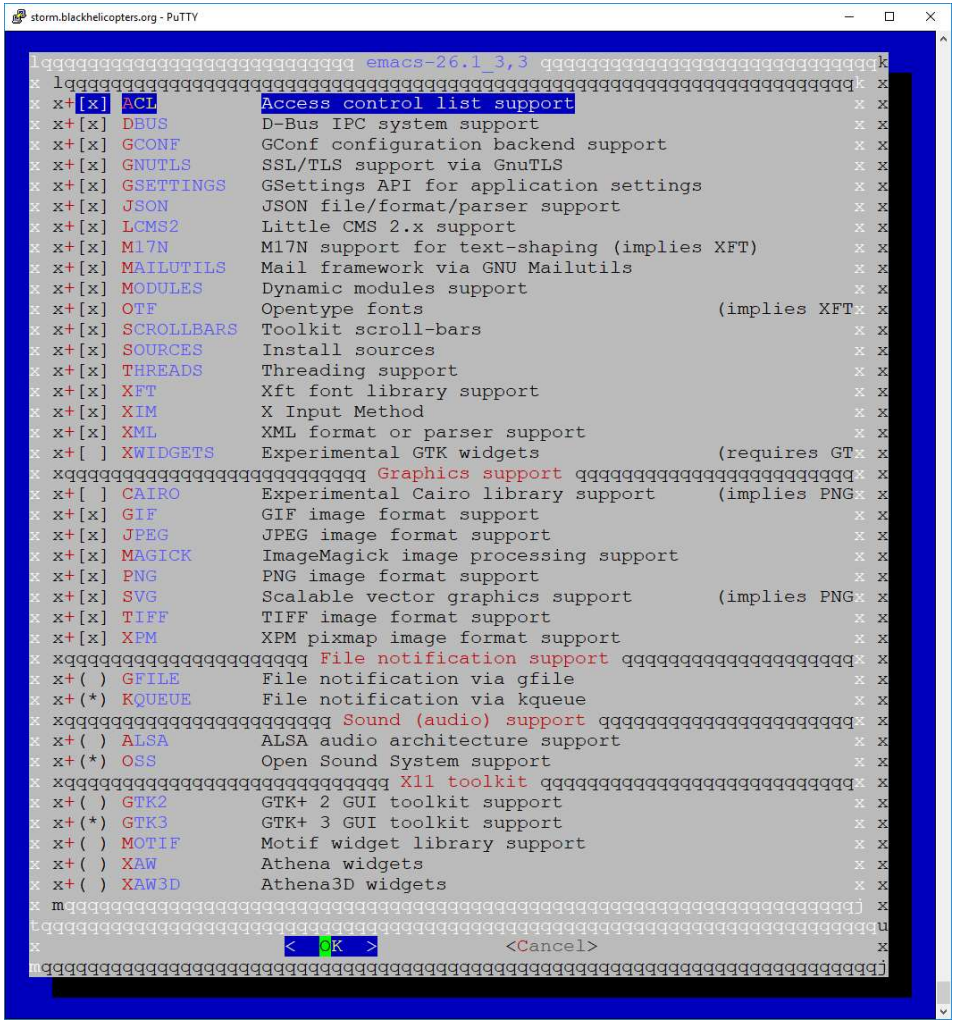
- FreeBSD is deliberately small
- Complete base install with source under 1 GB
- Packages include sudo, vim, (ships with nvi), X.org, GnuPG, nmap, Emacs, LDAP, modern mail clients…
- It’s your system. Install only what you need.

Package Tools

- Got rid of old pkg_tools a few years back, replaced with pkg(8)
  - pkg install emacs-nox
  - pkg upgrade
  - pkg search vim
  - pkg autoremove
  - pkg lock
- /usr/sbin/pkg bootstraps pkg package, allowing tooling to stay synced with packages
- Includes search and query language for automation

Package Source Code

- FreeBSD ships with the source for building packages, called “ports”
- /usr/ports contains ~33,000 ports
- Many have variants and flavors: with and without LDAP, X, etc
- /usr/ports/editors/emacs has 27 options
- Flags for licensing, redistribution, etc
- Checksums to catch file damage


storm.blackhelicopters.org - PuTTY
emacs-26.1_3,3
[x] ACL Access control list support
[x] DBUS D-Bus IPC system support
[x] GCONF GConf configuration backend support
[x] GNUTLS SSL/TLS support via GnuTLS
[x] GSETTINGS GSettings API for application settings
[x] JSON JSON file/format/parser support
[x] LCMS2 Little CMS 2.x support
[x] M17N M17N support for text-shaping (implies XFT)
[x] MAILUTILS Mail framework via GNU Mailutils
[x] MODULES Dynamic modules support
[x] OTF Opentype fonts (implies XFT
[x] SCROLLBARS Toolkit scroll-bars
[x] SOURCES Install sources
[x] THREADS Threading support
[x] XFT Xft font library support
[x] XIM X Input Method
[x] XML XML format or parser support
[ ] XWIDGETS Experimental GTK widgets (requires GT
Graphics support
[ ] CAIRO Experimental Cairo library support (implies PNG
[x] GIF GIF image format support
[x] JPEG JPEG image format support
[x] MAGICK ImageMagick image processing support
[x] PNG PNG image format support
[x] SVG Scalable vector graphics support (implies PNG
[x] TIFF TIFF image format support
[x] XPM XPM pixmap image format support
File notification support
( ) GFILE File notification via gfile
(*) KQUEUE File notification via kqueue
Sound (audio) support
( ) ALSA ALSA audio architecture support
(*) OSS Open Sound System support
X11 toolkit
( ) GTK2 GTK+ 2 GUI toolkit support
(*) GTK3 GTK+ 3 GUI toolkit support
( ) MOTIF Motif widget library support
( ) XAW Athena widgets
( ) XAW3D Athena3D widgets
< OK >
<Cancel>

Poudriere mass package builder

- Configure your packages and build once, install everywhere
- Isolates build from host userland through lightweight virtualization
- Can build for host release, or any earlier supported release
- Can build i386 binaries on amd64 hardware
- Possible to build whole FreeBSD ports tree as one process
- Offer clients packages by HTTP, NFS, whatever
- Can use memory file systems for build space
- Can ccache

Patches and Upgrades

- Supports binary upgrades via freebsd-update(8)

  # freebsd-update fetch install

  # freebsd-update upgrade –r 12.0-RELEASE

- Ports tree has its own utility, portsnap(8)

  # portsnap auto – fetch if not present, update if present

- Package updates via pkg

Running Binaries from the Wrong OS

- A kernel can have multiple ABIs
- Primordial BSD programs had a *brand* to say “this binary uses this kernel ABI.”
- FreeBSD doesn’t emulate the Linux ABI; it implements it
- Currently at 2.6.32, about due for a rev
- Userlands for CentOS 6 & 7 in ports, runs in a chroot
- Add-ons include linprocfs and linsysfs
- Dropped support for OSF/1, SCO, and SVR4

Jails

- Lightweight virtualization
- Uses host kernel with isolated process context, networking, and filesystem space to give illusion of standalone system
- Users can identify they’re in a jail, but can’t escape
- Can run multi-interface jails without any processes as firewalls
- Manage jails from the host or via ssh
- Host runs only essential features

Jail Features

- Utilize ZFS clones to create multiple similar jails
- Single OS image plus loopback mounts to create multiple similar jails that upgrade simultaneously
- Runs any FreeBSD older than the host, back to 4.X
- Hierarchical
- Can control which jails get what filesystems, and who can mount
- Shared SYSVIPC space, or no?
- Run Linux in a jail
- Resource controls with rctl(8)

Configuring jails: /etc/jail.conf

```
$j="/jail";
path="$j/$name";
host.hostname="$name.mwl.io";
exec.consolelog="/var/tmp/$name";

loghost {
  ip4.addr="203.0.113.231";
}

logdb {
  ip4.addr="203.0.113.232";
}
```

Blacklistd

- Scanners are a pain
- Fail2ban and other solutions ease that pain the hard way
- libblacklistd gives programs a way to declare “ban this IP for X time”
- Whitelist admin IPs, everything else can be blocked
- Uses any of the three FreeBSD firewalls

Three firewalls? Wait? What?

- No, not like firewalld and firewall-config and whatall
- These are packet filters
- PF – the current standard, ported and forked from OpenBSD
- IPFW – original packet filter
- IPF – cross-platform packet filter

- Use PF

Example PF ruleset

```
ext_if="vtnet0"
set skip on lo

scrub in
anchor "blacklistd/*" in on $ext_if
block in
pass out

pass in on $ext_if proto tcp to ($ext_if) port {22, 53, 80, 443}
pass in on $ext_if proto udp to ($ext_if) port {53, 33433 >< 33626}
pass in on $ext_if proto {icmp, icmp6}
```

Libarchive

- “I want to write a new FreeBSD installer and package manager, but need to sanely open distribution files. And zip files. And, and, and…”
- Thus born libarchive, general-purpose archive program
- Reads many different tar archives, ISO, zip, ar, MS CAB, LHA, mtree, RAR, XAR, gzip, bzip, xz, lzip, compress, uuencode…
- Writes ustar, pax interchange format, octet-oriented cpio, zip, two different shar, ISO, 7-zip, ar, XAR…

```
# tar –xf /home/mwl/random.iso
# tar –cf backup.iso /home/mwl/
```

BSD is Self-Hosting

- All BSDs ship with the OS source code, and the tools to build it
- Don’t want binary updates? Want to strip parts out?

```
# cd /usr/src
# make update && make buildworld && make kernel
# reboot
# cd /usr/src && make installworld
```

- Set your local options in /etc/src.conf

```
WITHOUT_GNU=YES
WITHOUT_SENDMAIL=YES
WITHOUT_CLANG=YES
```

Debugging and Developing

- Ships with everything you need to develop and build the OS
- Easy to enable crash dumps and trackbacks
- Variety of crash dump formats, from simple tracebacks to full memory dumps
- In /etc/rc.conf, set

  dumpdev="AUTO"
- Start with default method, can increase as needed

Documentation

- FreeBSD is self-documenting
- Everything has a man page
- If it doesn’t, it’s a bug.
- All FreeBSD resources under freebsd.org
  - docs.freebsd.org
  - forums.freebsd.org
  - docs.freebsd.org/mail
  - bugs.freebsd.org

Handbook and FAQ

- FreeBSD Handbook: long-form answers to introductory questions
- FreeBSD FAQ: short-form answers

Development

- Most discussion on mailing lists
- reviews.freebsd.org contains patches & discussion
- Main source code repo is subversion
- Constant automatic build testing
- Ports have experimental runs for stuff like X, OpenSSL, KDE upgrades

Community

- Ultimately, FreeBSD is people
  - Kernel developers
  - Userland developers
  - Integrators of third party software
  - Docs writers
  - Advocates
  - Podcasters
  - Tech Support
  - Enthusiasts

Questions?

- But before I answer…

Dear IRS: Tonight Is Now *Tax-Deductible*

Questions?

- </commercial>