

NetBSD 2018



AsiaBSDCon 2018 quickie

Who am I?

- Makoto FUJIWARA
 - mef @ NetBSD.org
 - (makoto @ ki.nu)
 - makoto @ if.t.u-tokyo.ac.jp
 - became a NetBSD developer in 2010
 - member of the TNF board (since last fall)
- Working area
 - NetBSD/pkgsrc

http://www.ki.nu/~makoto/pkgsrc/check-update/00_Summary.html

Summary

- NetBSD 8.0 coming soon
- GSoC 2018
- Support for new architectures
- Hardware Support
- Status of Meltdown and Spectre
- Security Topics
- VCS
- Debugging facilities and LLDB
- LLVM sanitizers and kernel fuzzing
- pkgsrc
- MP-safe Network Stack

NetBSD 8.0 coming soon

Release Engineering (Releng) team is working on it

NetBSD 6.0 2012-10-17

NetBSD 7.0 2015-09-25

NetBSD 8.0 2018

- - - NetBSD is 25 years old now - - -

GSoC 2018

NetBSD taking part in Google Summer of Code 2018

<http://mail-index.netbsd.org/current-users/2018/02/12/msg033208.html>

Support for new architectures

- aarch64 (see next slide)
- G5 support in NetBSD/macppc (macallan@)
- Atari & Milan support (tsutsuii@)
- Cobalt image revival (tsutsuii@)
- Support for **MANY** arm boards (jmcneill@)

Aarch64 (arm64)

Work by:

- Matt Thomas (matt@)
- Ryo Shimizu (ryo@)
- Toru Nishimura (nisimura@)
- Nick Hudson (skrll@)
- Jared McNeill (jmcneill@)

Support:

- Multiuser Mode
- FDT
- SMP and some other features are not yet available...

Will be committed soon

Hardware Support

- RaSCSI
 - SCSI device emulator over GPIO for the Pi
 - Good for Retro Computer, X68000 etc
 - It will work on linux/rpi and NetBSD/rpi
- Upcoming PCI adapter for DNARD (flxd@)
- Many fixes and improvements in many drivers

Status of Meltdown and Spectre

- Spectre (V1)
 - No fix yet
- Spectre (V2)
 - Waiting for Intel microcodes...
 - At the same time, looking at retpoline
- Meltdown (V3)
 - Fixed with **SVS**, on x86_64 (amd64) only

Security Topics (maxv@)

- Stronger Kernel Mappings (2016)
 - W^X
 - Reorganize several areas to mitigate certain possible attacks
- Stronger amd64/i386/xen Ports (2016-2017)
 - A lot of fixes to improve safety (and performance while here)
- Kernel ASLR on amd64 (2017)
 - As of today, the strongest implementation of KASLR in the world
- Audit of the Network Stack (2018)
 - Dozens of bugs and vulnerabilities identified
 - Critical fixes also sent to FreeBSD, OpenBSD

VCS (version control system)

- Looking for CVS replacement
- Mercurial server running as a pivot (joerg@)
- Git repository at IJ and NetBSD now

Debugging facilities - ptrace(2) (kamil@)

- Improvements

- Synchronize Machine-Independent features with Linux and FreeBSD
- Coverage of majority of the features with regression tests (almost 1000 ATF tests executed for x86)

- WIP

- Improve correctness of handling: signals, forks/vforks, threads (LWPs)
- Cover remaining behavior assumed by debuggers with new ATF tests

- TODO

- Machine-Dependent features (like XSAVE, XSAVEOPT for NetBSD/amd64)
- Research new syscall implementing StopTheWorld() for Leak Sanitizer (?)

The LLVM debugger (kamil@)

- LLDB porting to NetBSD/amd64
 - Use modern process plugin framework with remote debugging facilities
 - Initial support for tracing programs with a single thread
 - Read NetBSD core files of programs with multiple threads
 - NetBSD/amd64 buildbot in the LLVM farm building Clang+LLVM+LLDB
- WIP
 - Enhance correctness, tracing processes with multiple threads, integration with the base system
- TODO
 - i386 and non-x86 porting
 - kgdb(4) frontend, reading kernel core files
 - ipkdb(4) frontend (?)

Code quality improvement (kamil@)

- LLVM Sanitizers and sibling features
 - Done [usable with bugs]: **Undefined Behavior Sanitizer** (all ports), **Address Sanitizer** (amd64, i386), **Thread Sanitizer** (amd64), **Memory Sanitizer** (amd64), **libFuzzer** (tested on amd64), **SafeStack** (tested on amd64), **Thread Sanitizer in the Go runtime** (amd64)
 - WIP [incomplete]: **X-Ray** (function call tracing system with instrumentation), **Scudo** (hardened allocator), **Leak Sanitizer** (amd64), **Memory Sanitizer in the Go runtime**
 - TODO: **kernel sanitizers** (KUBSan/KASan) [GSoC proposals], **sanitizers in GCC** (mostly sync the code from the LLVM sanitizers), **integration** of the sanitizers and sibling features **with the NetBSD basesystem distribution** [GSoC proposals]

Code quality improvement (kamil@)

- Syzkaller - unsupervised, coverage-guided kernel fuzzer
 - Initial work done by Dmitry Vyukov (Google), Christos Zoulas (NetBSD), Utarsh Anand (NetBSD)
 - The first kernel bug detected and fix applied (“mmap() a file, PaX MPROTECT can produce an unkillable process”
<http://gnats.netbsd.org/52658>)
 - Resume porting [GSoC proposal]
- Kernel fuzzing TODO
 - KCOV-like coverage, automation with Anita, ship tools in pkgsrc, i386 support, teach Syzkaller utilities about NetBSD (symbolizers of kernel crash cores, reproducer generators, etc)...
 - rumpkernel fuzzing (?)

pkgsrc

- pkgsrcCon 2018 in Berlin (6th to 8th of July)
- Tor pkgsrc repo (alnsn@)
- Postfix support for blacklistd (christos@)
- Fixing bugs so that NetBSD/i386 & NetBSD/amd64 are supported in Golang v1.10 (christos@)
- Go14 and go work on NetBSD/arm (oshimaya@)

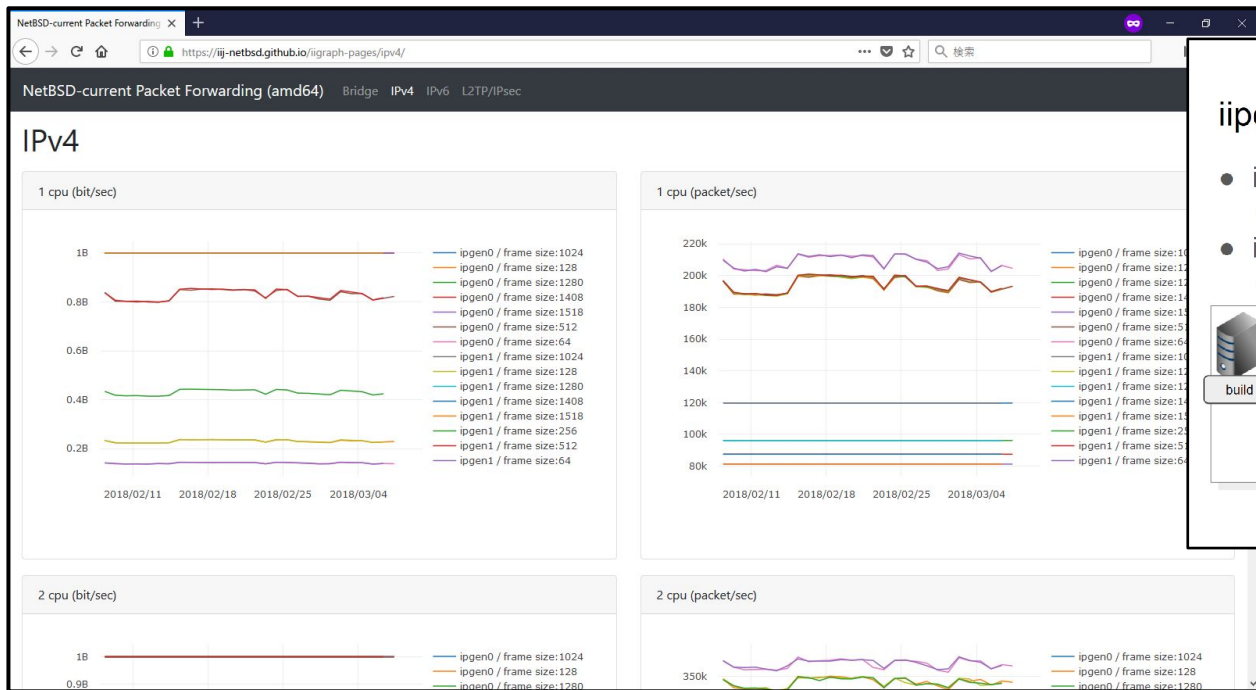
MP-safe Network Stack (as of 8.0)

- Device drivers
 - vioif, vmx, wm, ixg, ixv
- Layer 2
 - Ethernet, bridge, fast forward
- Layer 3 (mostly)
- Interfaces
 - gif, ipsecif, l2tp, pppoe, vlan
- Others
 - bpf, ipsec, npf, opencrypto, pfil

Ready for use for routers, but disabled by default.

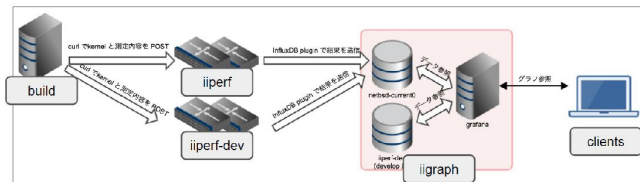
You can try by adding `NET_MPSAFE` option to your kernel.

Daily Benchmarks of Network Performance



iiperf and iigraph

- iiperf
 - Performance measurement
- iigraph
 - Datastore and visualization



Developed and deployed by
s-yamaguchi@IIJ and suzu-ken@IIJ

25

<https://ij-netbsd.github.io/iigraph-pages/>

