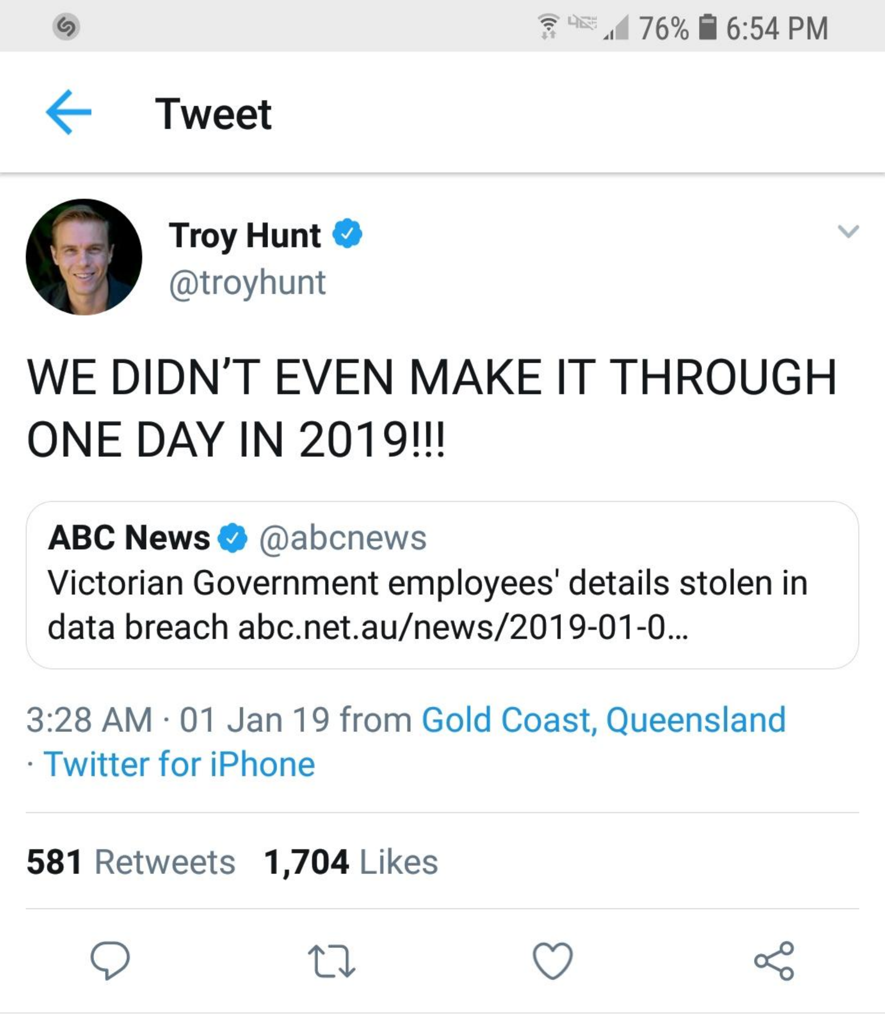
Getting -current with OpenBSD

NYC*BUG 2019-01-02



Troy Hunt
@troyhunt

WE DIDN'T EVEN MAKE IT THROUGH ONE DAY IN 2019!!!

ABC News @abcnews
Victorian Government employees' details stolen in data breach abc.net.au/news/2019-01-0…

3:28 AM · 01 Jan 19 from Gold Coast, Queensland · Twitter for iPhone

581 Retweets **1,704** Likes

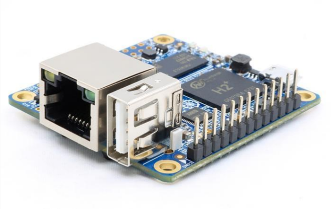
Things to talk about

- Hardware: armv7, arm64
- Software: pledge(2), unveil(2)
- Toolchains
- Meltdown/Spectre/SMT
- RETGUARD
- Ports land
- Playing Steam games on OpenBSD!!!

Hardware

armv7

- 6.0
  - armv7 platform is born
- 6.1
  - Bye bye armish, zaurus (pre-armv7)
  - PIE on arm
- 6.3
  - Clang
  - VFP/NEON instructions
- -current
  - ld.lld



arm64

- 6.1
  - arm64 platform is born
  - 100% LLVM toolchain (a first!)
- 6.3
  - SMP
- 6.4
  - ACPI
  - radeondrm(4)

OpenBSD/arm*

- Installation
  - Still over serial line for a lot of things
    - But fortunately one cable can be used on many things
- Using it
  - armv7
    - No graphics (yet)
    - No SMP (yet)
  - arm64
    - Graphics with radeondrm(4)
    - SMP
- Packages (>8300)

Software

pledge(2) and unveil(2)

- pledge(2): 5.9
    - tame(2)
    - Works on syscalls
- unveil(2): 6.4
    - Works on filesystems

pledge(2)

- #include <unistd.h>
- int pledge(const char *promises, const char *execpromises);

pledge(2)

- Using
  - Error condition is -1
  - if (pledge(“”, NULL) == -1)
  - You can call pledge multiple times over the life of a program
    - “Fine tuning” pledge
- **It is the responsibility of the programmer to know what syscalls will be used.**
  - Sometimes this might be a little unusual at first
    - For example, using sndio would require the following promises string for sio_open
      - stdio rpath wpath cpath inet unix dns audio
    - You can drop all but audio after the final sio_open call

pledge(2)

- Things that are pledged (outside base)
  - Chromium
  - Firefox
  - glib2
  - gtk+3/gtk+4
- Languages that have pledge support (outside base)
  - Haskell
  - Go
  - Lua
  - Node

unveil(2)

- #include <unistd.h>
- int unveil(const char *path, const char *permissions);

unveil(2)

- Using
  - Error condition is -1
  - unveil is a pledge promise!
  - if (pledge(“… unveil …”, NULL) == -1)
  - if (unveil(“/home/brian”, “r”) == -1)
  - Further calls **add** to the list (opposite of pledge!)
- Permissions
  - “r” (read, think pledge rpath promise)
  - “w” (write, think pledge wpath promise)
  - “x” (execute, think pledge exec promise)
  - “c” (create, think pledge cpath promise)

unveil(2)

- Chrome/Iridium already use unveil!
  - Can be disabled with --disable-unveil if you encounter bugs

Toolchains

- gcc-4.2.1 → clang-6.0.0
  - amd64
  - arm64
  - armv7
  - i386
  - mips64/mips64el (not default compiler)
  - sparc64 (not default compiler)
- GNU ld.bfd 2.17 → LLVM ld.lld 6.0.0
  - amd64
  - arm64
  - armv7
  - i386

Meltdown/Spectre/SMT

- -mretpoline
- hw.smt sysctl

RETGUARD

- Per-function stack cookie
- Implemented as an independent pass in LLVM

Ports land

- Toolchain excitement
  - Especially ld.lld
- Exotics arch support
- The 10,000 packages milestone

The OpenBSD Gaming Project

- Mono
- Fnaify
- Open source gaming engines
- PlayOnBSD.com
- Demo?

Other things of interest

- macppc64 port?
- LLVM 7.0.1 forthcoming
  - OpenPOWER?
  - RISC-V?

Thanks!

Bother me:

bcallah@openbsd.org

@__briancallahan (Twitter)

@bcallah@bsd.network (Mastodon)