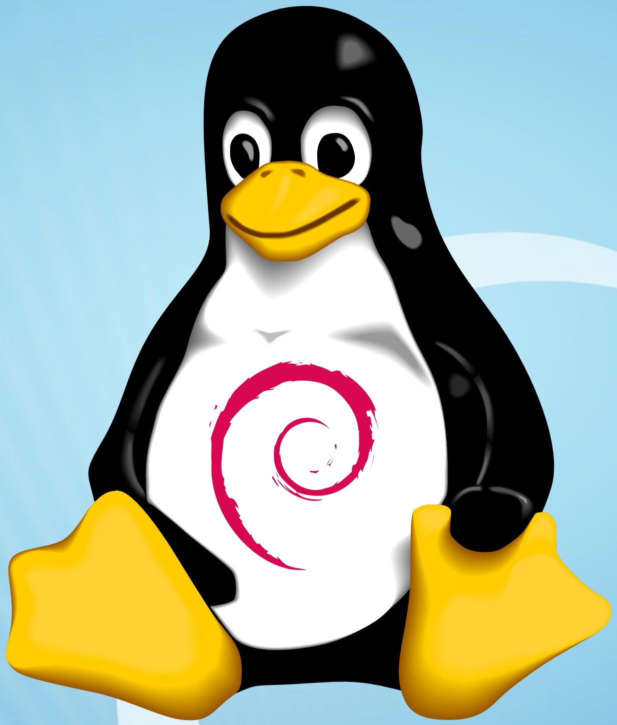
What's new in the Linux kernel

and what's missing in Debian



Ben Hutchings · DebConf 18

Ben Hutchings

- Working on Linux kernel and related code for Debian and in paid jobs for over 10 years
- Debian kernel and LTS team member, doing about half the kernel packaging work
- Maintaining Linux 3.16.y stable update series on kernel.org
- Maintaining CIP Linux 4.4.y stable branch

Linux releases early and often

- Linux is released about 5 times a year (plus stable updates every week or two)
- ...though some features aren't ready to use when they first appear in a release
- Since my talk at mini-DebConf Cambridge, Linus has made 3 releases (4.15-4.17) and 4.18 is imminent
- Good news: we have lots of new kernel features coming to testing/unstable
- Bad news: some of them will need changes elsewhere to be useful

Recap of last year's features

- schedutil: still not the default cpufreq governor
- Zoned recording: dm-zoned-tools still at RFP: #882640
- Block layer replacement: changed default for SCSI; MMC switched over completely; md-RAID still TBD
- ARM graphics drivers: Mali kernel driver under GPL and packaged in contrib; user-space still proprietary
- statx(): supported by glibc 2.28; few applications using it

RISC-V [4.15-ongoing]

- New ISA defined since 2010, maintained by industry consortium
- Architecture is open (public docs, no licence fees), but implementations may be proprietary and include vendor extensions
- Scalable from 32-bit microcontrollers to 64-bit high-performance processors (with room to extend to 128-bit)
  - Many features optional, including multiplication/division and MMU
  - Common feature-set defined for general-purpose OSes like Linux
- Basic support included in Linux 4.15; more features and drivers added since
- Debian port: riscv64

Security hardening [ongoing]

- `struct timer_list` (used to track time-outs, delayed work, etc.)
  - Used to have function pointer and argument for the function (unsigned long)—very useful to an attacker who can overwrite the structure
  - Now function is always called with a pointer to the structure, so attacker can't easily control the argument
- Usercopy (copying data between user and kernel memory)
  - Already had range checks to prevent copies overflowing stack or heap area
  - Known limitation: didn't catch overflows within the same memory area, so a bug could still lead to overwriting other parts of a structure
  - Private heaps (kmem_cache) can now have a “whitelist” that defines which parts of each allocation—if any—usercopy should be allowed in

Speculation leak mitigation [ongoing] - 1

- Speculative execution allows CPUs to avoid waiting for slow operations like memory reads by predicting the result
- Results of speculation are buffered until the prediction is checked, and discarded if it was wrong
- Researchers discovered that misprediction can allow speculative execution to bypass hardware or software access control, or jump to attacker-controlled address
- Even though results are discarded, changes to memory caches are not undone and can leak sensitive information to an attacker
- Fixing this may require big changes to CPU design—but microcode and software can mitigate

Speculation leak mitigation [ongoing] - 2

- Spectre v1 (CVE-2017-5753): bounds check bypass
  - Most CPUs on most release architectures affected
  - Mitigated by masking value after check
    - Small performance cost, if applied selectively
    - All sensitive checks need to be identified and patched; ongoing effort
- Spectre v2 (CVE-2017-5715): branch target injection
  - Most CPUs on most release architectures affected
  - Mitigated on x86, Power and System z by disabling or defeating indirect branch prediction in the kernel
  - Additionally mitigated on x86 using new microcoded features

Speculation leak mitigation [ongoing] - 3

- Meltdown (CVE-2017-5754): rogue data cache load
  - Intel x86, some ARM-architecture, IBM POWER CPUs affected
  - Mitigated by Page Table Isolation or cache flush
    - Slows down system calls and interrupt handling
    - Not yet done for i386
- Spectre-NG v4 (CVE-2018-3639): speculative store bypass
  - Most x86 and some ARM-architecture CPUs affected
  - Mostly mitigated by same software changes as v1 and v2
  - Additionally mitigated on x86 using new microcoded features

Speculation leak mitigation [ongoing] - 4

- CVE-2018-3665: floating-point/vector register leak
  - Only Intel x86 CPUs affected, and only if OS uses “lazyfpu”
  - Linux already disabled lazyfpu on recent CPUs with XSAVEOPT
  - Mitigated by disabling lazyfpu completely
- Spectre v1.1 (CVE-2018-3693): bounds check bypass store
  - Affected CPUs and mitigations similar to v1
- Spectre v1.2: read-only protection bypass
  - Affected CPUs similar to Meltdown
  - Appears to make v1.1 more powerful; not a security issue in itself?

Y2038 [4.18-ongoing]

- Kernel internal interfaces updated to use 64-bit time types in all configurations
- 32-bit kernel configurations can now include 64-bit versions of most time-related system calls
- Not yet enabled by any architecture!
- glibc doesn't support both 32-bit and 64-bit time_t at the same time, and review of the necessary changes is going slowly
- Will miss buster, but probably be ready for bookworm
- Could dpkg-buildflags enable LFS and 64-bit time by default?

FUSE in user namespaces [4.18]

- Any user can create a user namespace (userns) and be the root user in their own little world
  - Disabled in Debian by default, because this exposes a *lot* of security bugs
- Most Linux filesystems are not robust against maliciously constructed disk images
- mount(2) is restricted, so you can't use this to attack filesystem code
- FUSE (filesystem in user-space) moves the security problem out of the kernel
- FUSE now considered robust enough to be mounted in any userns
- Any filesystem can be implemented through FUSE...in theory
- Do we need to package more FUSE filesystems?

SATA link power management [4.15]

- SATA: high speed serial link to storage devices; draws significant power even when no data being transferred
- Link power management (LPM) can switch into lower power modes when idle
  - Recent Intel processors may also enter lower power state when this happens
- “Aggressive LPM” gives high power savings, but risks data loss due to hardware bugs—so not enabled in Linux
- Linux can now set LPM settings similar to Windows on Intel-based laptops—saves more power and believed to be well-tested
- Enabled in Debian kernel from 4.17; but seems to cause boot hangs on some systems—may be necessary to blacklist some drives

Packaging changes

- Template source package for code signing
- More flexible selection of binary packages, to support derivatives and backports that don't want them all
- Kernel config files moved into new binary packages (linux-config-*version*)
- Removed remaining dependencies on Python 2
- Preparation for armhf and arm64 packages with PREEMPT_RT
- Moved all repositories to Salsa—merge requests welcome!

Questions?

Credits & License

- Content by Ben Hutchings
  www.decadent.org.uk/ben/talks/
  License: GPL-2+
- Original OpenOffice.org template by Raphaël Hertzog
  raphaelhertzog.com/go/ooo-template
  License: GPL-2+
- Background based on “Serenity” theme by Edward Padilla
  wiki.debian.org/DebianArt/Themes/serenity
  License: GPL-2