The state of the Linux kernel security

Dmitry Vyukov (dvyukov@)

OpenSSF: Securing Critical Projects WG (Sep 24, 2020)

Disclaimer

Based on my personal observations over the past 5 years being involved in the Linux kernel security area.

My background

Dynamic Tools team at Google since 2011:

- **Sanitizers** (ASAN, TSAN, MSAN, ...)
- **Hardening/production** (CFI, GWP-ASAN, MTE)
- **Fuzzing** (LibFuzzer, go-fuzz, OSS-Fuzz, ...)

Kernel work since ~2015:

- **Sanitizers** (KASAN, KCSAN, KMSAN, ...)
- **Hardening/production** (memory initialization, KFENCE, MTE)
- **Fuzzing** (syzkaller, syzbot)

"Civilization runs on Linux" [1]

- Android (2B+ users)
- Cloud, HPC, servers
- ChromeBooks, notebooks, desktops
- IoT
- Cars
- Nuclear Submarines, Power Plants, Air/Car Traffic Control
- Large Hadron Collider, International Space Station
- ...
- Our coffee machines!

[1] from SLTS project which aims at maintaining kernel releases for 20+ years for industrial use

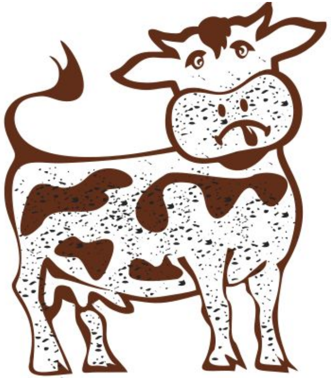
Security is Critical

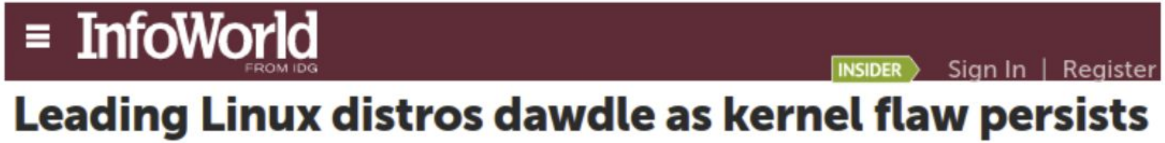
- Protects privacy of 2 billion people
- Protects corp, government information
- Protects safety-critical systems
- The first line of defence for:
  - all incoming network packets
  - untrusted apps
  - VM guests
  - USB/NFC/Bluetooth (inserting a USB clicker into your notebook)
- Cars/phones/plants: stability and safety are also critical

Linux kernel is **the most security-critical** infrastructure component in the world.

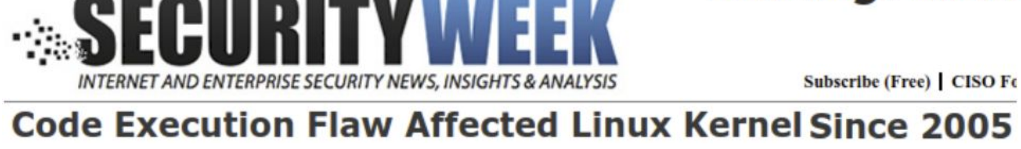
Tip of the Iceberg

Bugs with logos and bold headlines





Leading Linux distros dawdle as kernel flaw persists


SECURITYWEEK
INTERNET AND ENTERPRISE SECURITY NEWS, INSIGHTS & ANALYSIS
Subscribe (Free) | CISO Fo

Code Execution Flaw Affected Linux Kernel Since 2005

CVEs: the next layer of the iceberg*

Year/CVEs	Total	Code execution	Gain privileges
2017	453	169	125
2018	177	3	23
2019	170	5	19

*most are unaccounted, some are misaccounted

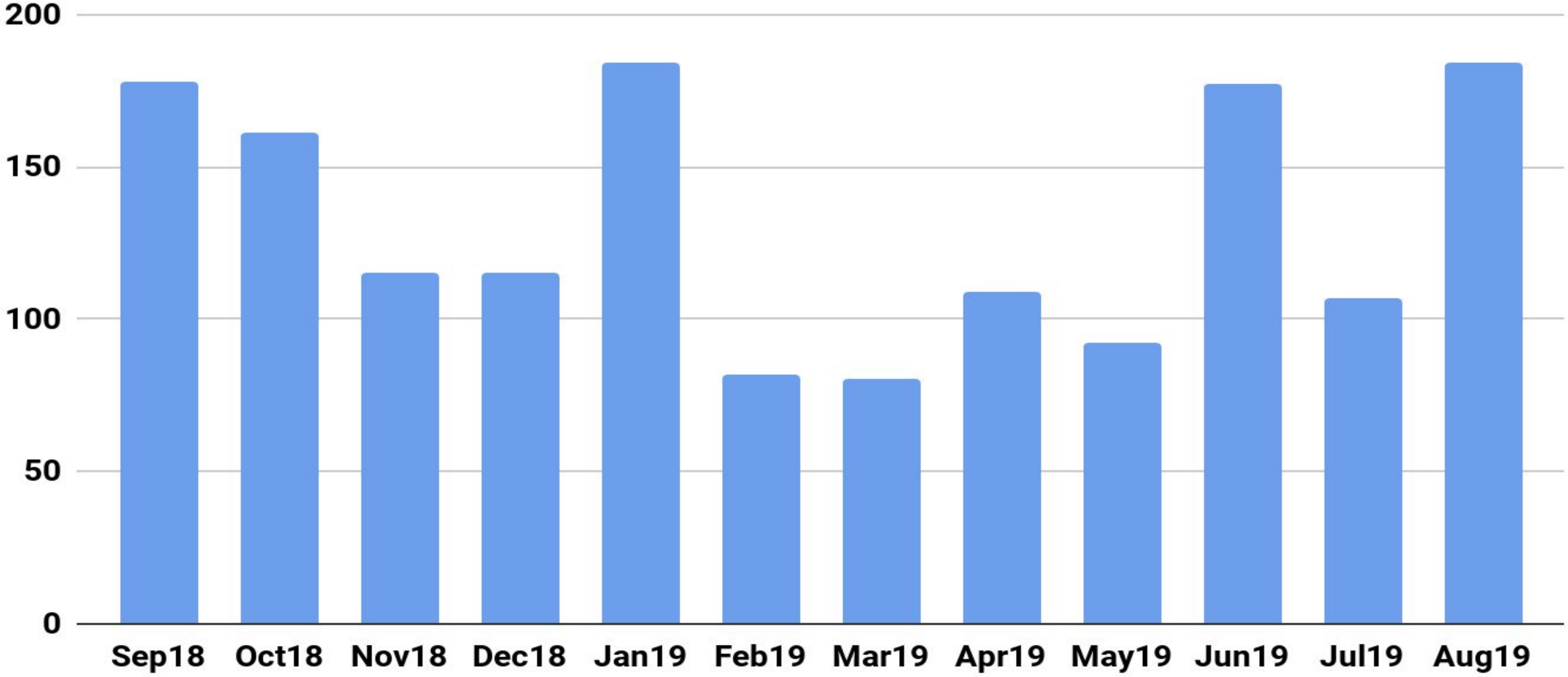
Part I: Bugs

syzbot: continuous kernel fuzzing

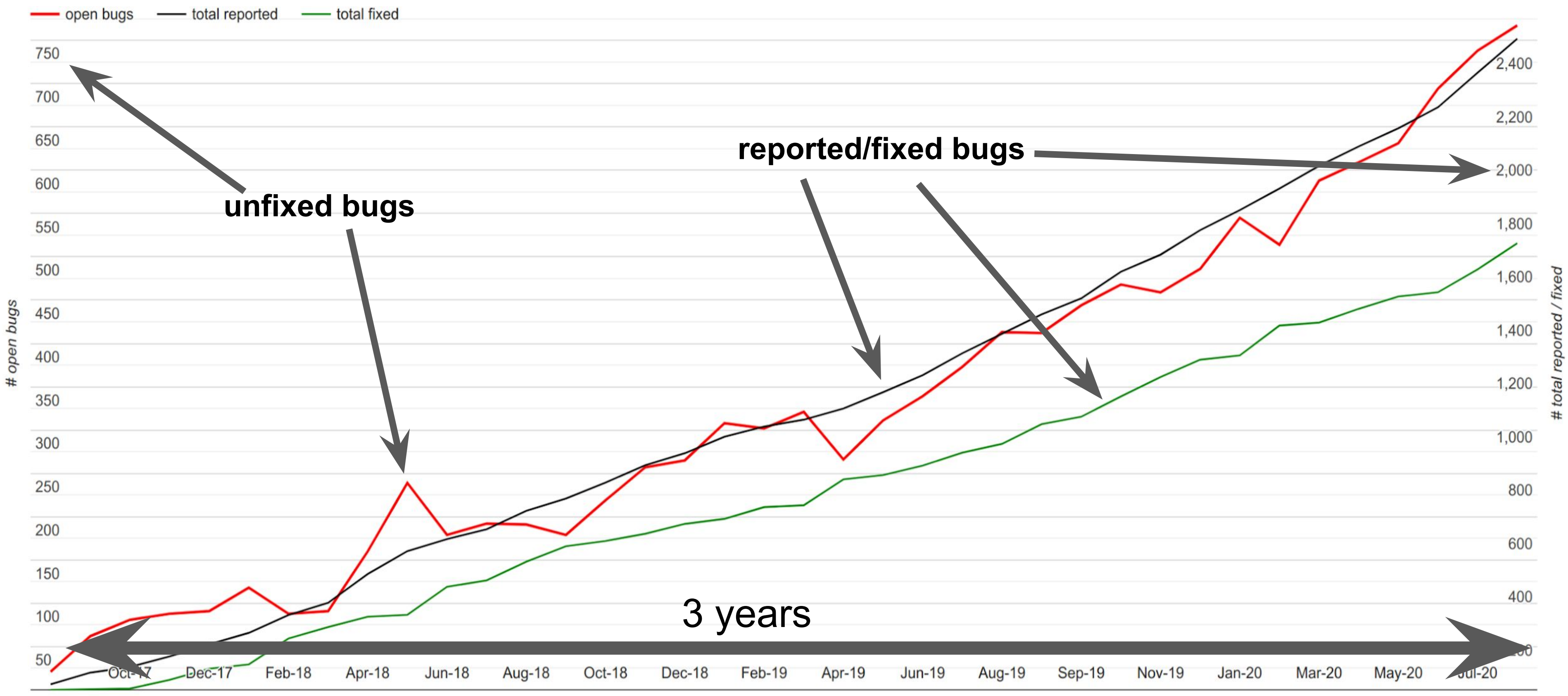
For 3 years ~200 bugs/month:

- 3900 bugs in upstream kernel
- ~same amount in Android/ChromeOS/internal kernels

syzbot bug rate



Upstream Bug Stats

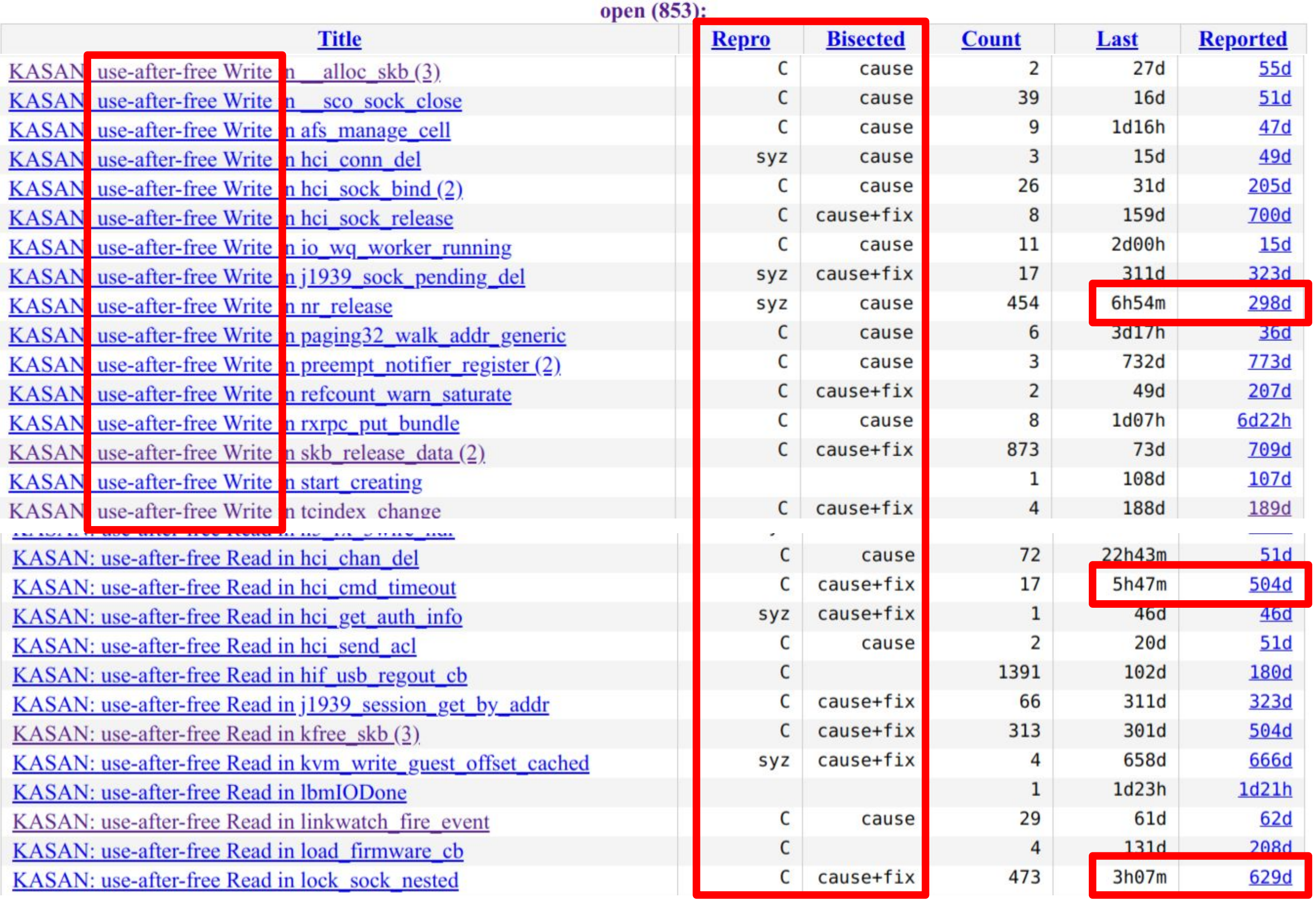


unfixed bugs

reported/fixed bugs

3 years

syzbot dashboard



open (853):

Title	Repro	Bisected	Count	Last	Reported
KASAN use-after-free Write n __alloc_skb (3)	C	cause	2	27d	55d
KASAN use-after-free Write n __sco_sock_close	C	cause	39	16d	51d
KASAN use-after-free Write n afs_manage_cell	C	cause	9	1d16h	47d
KASAN use-after-free Write n hci_conn_del	syz	cause	3	15d	49d
KASAN use-after-free Write n hci_sock_bind (2)	C	cause	26	31d	205d
KASAN use-after-free Write n hci_sock_release	C	cause+fix	8	159d	700d
KASAN use-after-free Write n io_wq_worker_running	C	cause	11	2d00h	15d
KASAN use-after-free Write n j1939_sock_pending_del	syz	cause+fix	17	311d	323d
KASAN use-after-free Write n nr_release	syz	cause	454	6h54m	298d
KASAN use-after-free Write n paging32_walk_addr_generic	C	cause	6	3d17h	36d
KASAN use-after-free Write n preempt_notifier_register (2)	C	cause	3	732d	773d
KASAN use-after-free Write n refcount_warn_saturate	C	cause+fix	2	49d	207d
KASAN use-after-free Write n rxrpc_put_bundle	C	cause	8	1d07h	6d22h
KASAN use-after-free Write n skb_release_data (2)	C	cause+fix	873	73d	709d
KASAN use-after-free Write n start_creating			1	108d	107d
KASAN use-after-free Write n tcindex_change	C	cause+fix	4	188d	189d
KASAN: use-after-free Read in hci_chan_del	C	cause	72	22h43m	51d
KASAN: use-after-free Read in hci_cmd_timeout	C	cause+fix	17	5h47m	504d
KASAN: use-after-free Read in hci_get_auth_info	syz	cause+fix	1	46d	46d
KASAN: use-after-free Read in hci_send_acl	C	cause	2	20d	51d
KASAN: use-after-free Read in hif_usb_regout_cb	C		1391	102d	180d
KASAN: use-after-free Read in j1939_session_get_by_addr	C	cause+fix	66	311d	323d
KASAN: use-after-free Read in kfree_skb (3)	C	cause+fix	313	301d	504d
KASAN: use-after-free Read in kvm_write_guest_offset_cached	syz	cause+fix	4	658d	666d
KASAN: use-after-free Read in lbmIODone			1	1d23h	1d21h
KASAN: use-after-free Read in linkwatch_fire_event	C	cause	29	61d	62d
KASAN: use-after-free Read in load_firmware_cb	C		4	131d	208d
KASAN: use-after-free Read in lock_sock_nested	C	cause+fix	473	3h07m	629d

Bugs/Security*

Use-after-free	18.5%
Heap-out-of-bounds	5.2%
Stack-out-of-bound	2.4%
Double-free	0.8%
Wild-access	4.8%
Uninit-memory	4.0%
GPF	20.2%
BUG/panic/div0	10.3%
deadlock/hang/stall	12.5%
WARNING	21.1%

* due to kernel nature lots of bugs have some security implications

** even "innocent" bug types can result in critical security issues

Bad stuff we found*

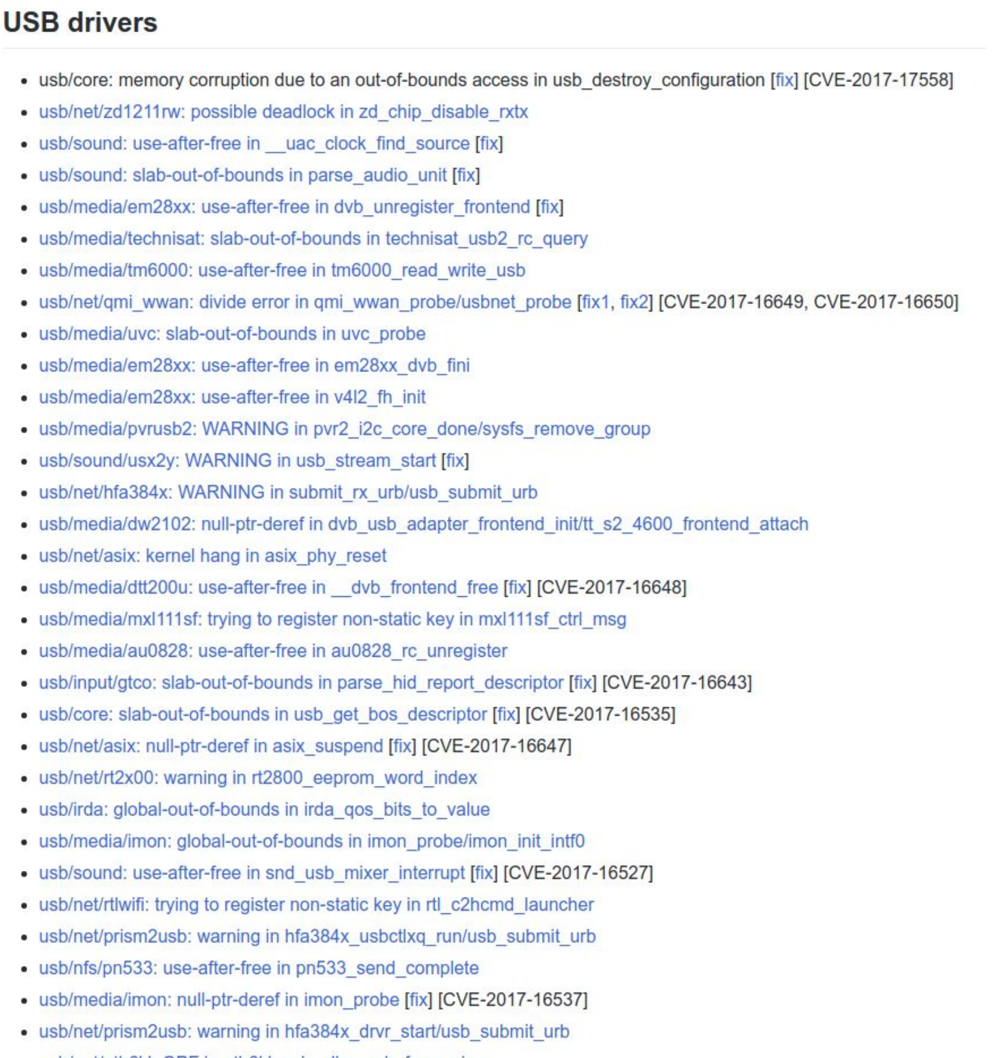
- Network remote
  - udp: properly support MSG_PEEK with truncated buffers
  - tcp: avoid infinite loop in tcp_splice_read()
  - ipv4: keep skb->dst around in presence of IP options
  - udp: on peeking bad csum, drop packets even if not at head *[manifested as hang]*
  - fou: prevent recursion in GUE error handler *[manifested as VM collapse]*
- VM escapes
  - kvm: use-after-free in complete_emulated_mmio
  - KVM: x86: Fix kernel info-leak in KVM_HC_CLOCK_PAIRING hypercall
  - kvm: fix page struct leak in handle_vmon *[manifested as memory leak]*
  - WARNING in __switch_to / WARNING in fpu__copy *[manifested as WARNING]*
- Local priv escalations/info leaks
  - Bad Binder: Android In-The-Wild Exploit
  - KASAN: use-after-free Read in binder_release_work
  - posix-timer: Properly check sigevent->sigev_notify
  - ... too many to enumerate ...

* <1% analyzed, few have CVEs

USB*

300+ bugs triggerable by any cable (code exec, info leaks, DoS) [1,2,3].

Limited/no coverage for lots of drivers.



USB drivers

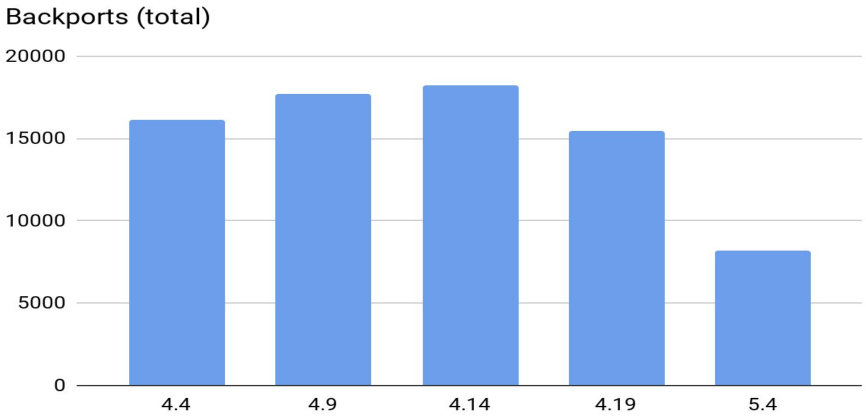
- usb/core: memory corruption due to an out-of-bounds access in usb_destroy_configuration [fix] [CVE-2017-17558]
- usb/net/zd1211rw: possible deadlock in zd_chip_disable_rxtx
- usb/sound: use-after-free in __uac_clock_find_source [fix]
- usb/sound: slab-out-of-bounds in parse_audio_unit [fix]
- usb/media/em28xx: use-after-free in dvb_unregister_frontend [fix]
- usb/media/technisat: slab-out-of-bounds in technisat_usb2_rc_query
- usb/media/tm6000: use-after-free in tm6000_read_write_usb
- usb/net/qmi_wwan: divide error in qmi_wwan_probe/usbnet_probe [fix1, fix2] [CVE-2017-16649, CVE-2017-16650]
- usb/media/uvc: slab-out-of-bounds in uvc_probe
- usb/media/em28xx: use-after-free in em28xx_dvb_fini
- usb/media/em28xx: use-after-free in v4l2_fh_init
- usb/media/pvrusb2: WARNING in pvr2_i2c_core_done/sysfs_remove_group
- usb/sound/usx2y: WARNING in usb_stream_start [fix]
- usb/net/hfa384x: WARNING in submit_rx_urb/usb_submit_urb
- usb/media/dw2102: null-ptr-deref in dvb_usb_adapter_frontend_init/tt_s2_4600_frontend_attach
- usb/net/asix: kernel hang in asix_phy_reset
- usb/media/dtt200u: use-after-free in __dvb_frontend_free [fix] [CVE-2017-16648]
- usb/media/mxl111sf: trying to register non-static key in mxl111sf_ctrl_msg
- usb/media/au0828: use-after-free in au0828_rc_unregister
- usb/input/gtco: slab-out-of-bounds in parse_hid_report_descriptor [fix] [CVE-2017-16643]
- usb/core: slab-out-of-bounds in usb_get_bos_descriptor [fix] [CVE-2017-16535]
- usb/net/asix: null-ptr-deref in asix_suspend [fix] [CVE-2017-16647]
- usb/net/rt2x00: warning in rt2800_eeprom_word_index
- usb/irda: global-out-of-bounds in irda_qos_bits_to_value
- usb/media/imon: global-out-of-bounds in imon_probe/imon_init_intf0
- usb/sound: use-after-free in snd_usb_mixer_interrupt [fix] [CVE-2017-16527]
- usb/net/rtlwifi: trying to register non-static key in rtl_c2hcmd_launcher
- usb/net/prism2usb: warning in hfa384x_usbctlxq_run/usb_submit_urb
- usb/nfs/pn533: use-after-free in pn533_send_complete
- usb/media/imon: null-ptr-deref in imon_probe [fix] [CVE-2017-16537]
- usb/net/prism2usb: warning in hfa384x_drvr_start/usb_submit_urb

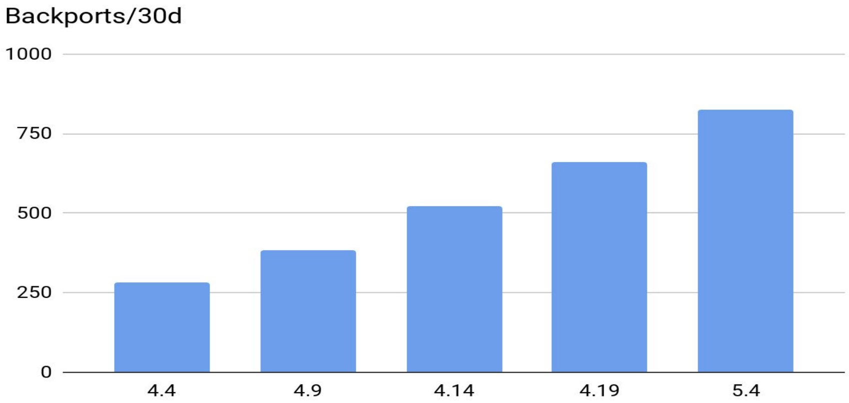
* USB is not special, similar for other subsystems

syzbot is not a complete solution

- <10% coverage
- only x86_64
- only VMs
- only "crashes"

Stable releases



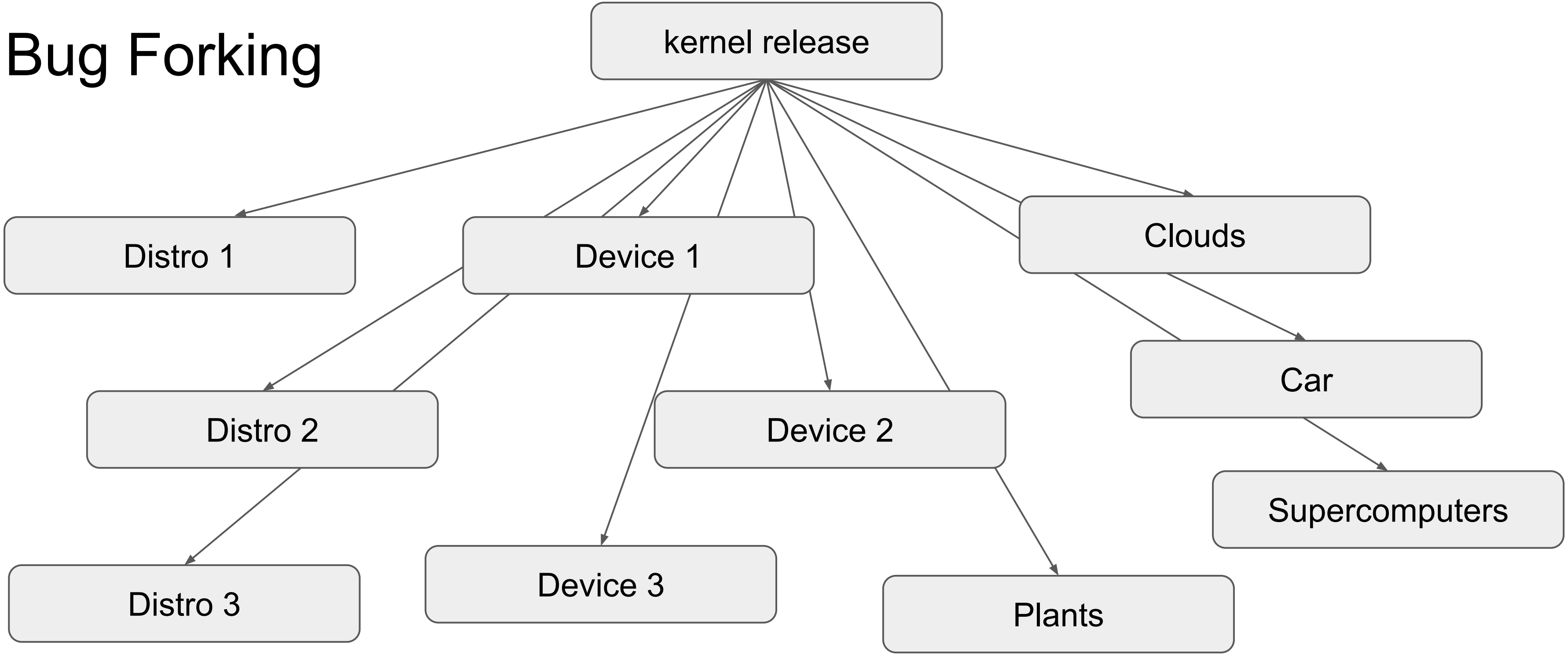


+ not backported fixes (700+)
+ not fixed upstream bugs (800+)
+ not found/detectable bugs (???)

>20'000 bugs/release

- not getting better
- introduced at the same rate
- these are "release-escaping" bugs

Bug Forking



Each bug fork is effectively a new bug for most practical purposes.

10K bugs x 10K forks = **100M bugs** 😢

Goal

Reduce bugs/release **100x**: 20'000 -> 200

Part II: Development Process

Lost patches

- 2-3% of patches are "lost"
- bias towards bugs fixes* (?)
  - fs: fix data race on mnt.mnt_flags (lost for **3 years**)
  - NFC: fix attrs checks in netlink interface (lost for **6 months**)
  - mm, oom: Introduce time limit for dump_tasks duration (lost for **2 years**)
  - ptrace: zero out siginfo_t in ptrace_peek_siginfo() (lost for **7 months**)
  - floppy: fix usercopy direction (lost for **5 months**)

* "public 0-days"

Lost bug reports

- Bugs are generally
  - fixed right away or...
  - get lost
- Re-reported bugs
  - net/xfrm: stack-out-of-bounds in xfrm_state_find (re-reported after 7 months)
  - general protection fault in kvm_ioapic_scan_entry (re-reported after 3 months)
  - BUG: unable to handle kernel paging request in vmx_vcpu_run (re-reported after 7 months)
  - KASAN: slab-out-of-bounds Write in process_preds (re-reported after 2 months)
  - WARNING: refcount bug in kvm_vm_ioctl (high-severity CVE after 3 months)

Bug tracking

- no triage/security assessment (generally)
- unmaintained subsystems
  - fbcon/vt: unmaintained, enabled in distros, full of bugs, accessible to users ("public 0-days")
  - NFC, EFS, HFS, LAPB, DECNET, USB_ZD1201 ("S: Orphan")
- "Nobody reads the kernel mailing list directly - there's just too much traffic"
- Fixed bugs in bugzilla are not closed
- Bugs are not expected to be in the bug tracker

Security practices in development

Ideally:

- tests
- fuzzers
- sanitizers
- coverage

Reality:

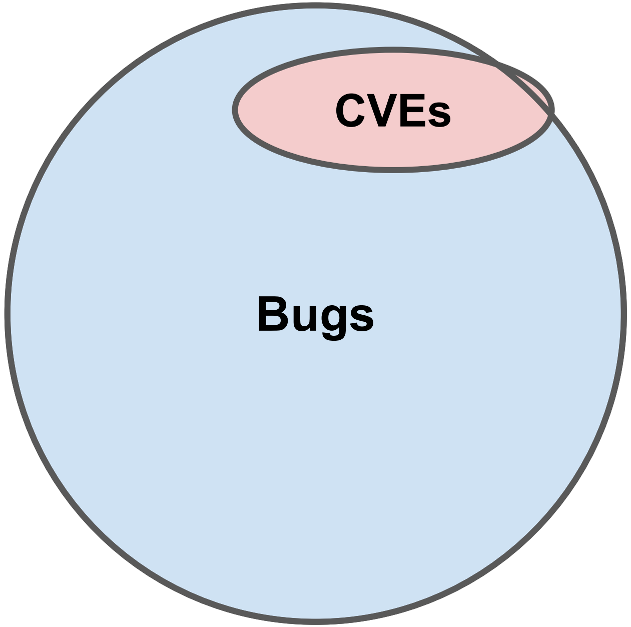
- new features w/o tests
- bug fixes w/o tests
- no fuzzers
- basic uses expose tool-detectable bugs
- test coverage unknown

"we know we don't have critical bugs" vs "we don't have known critical bugs"

Stable release process

- No consistent process to identifying backports
- Backports of security fixes take 6-12 months [1,2,3,...]
- "whole subsystems who still refuse to tag patches for stable"
- Patches dropped on merge conflicts (5700-1727=3973)
    - "FAILED patch + KASAN" = 127 "public 0-days"
- Stable backports
    - introduce use-after-free's (on network receive path)
    - break subsystems
    - break boot

Upstream / Distros Gap



Distros: "we fixed all CVEs" 😊

Upstream: "we don't do CVEs, use stable" 😊

Black hats: 😃

End user security: 😞

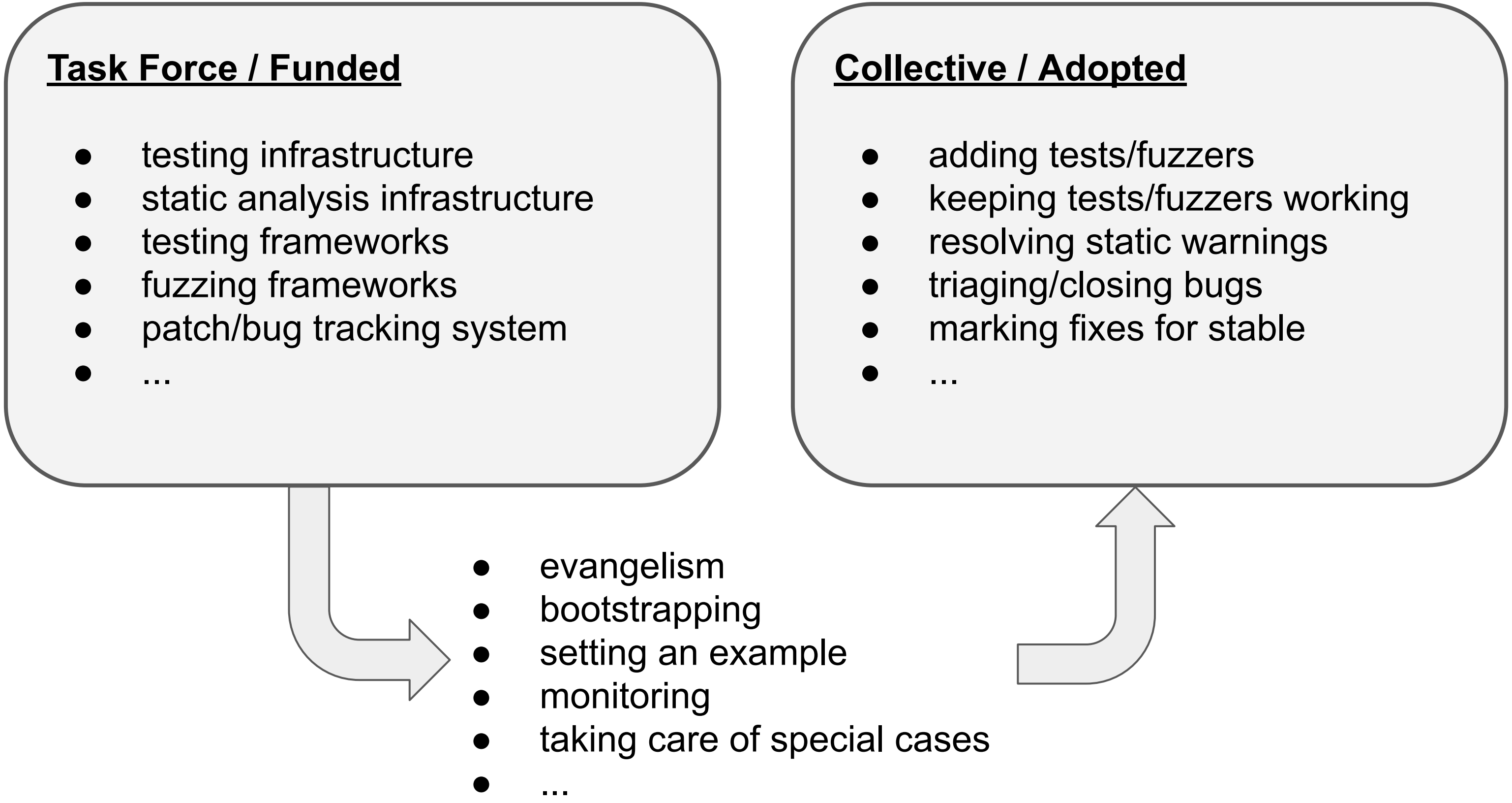
Stop issuing CVEs for kernel?

Action Plan

"What"

- reduce inflow of bugs
  - makes important processes feasible
- testing/fuzzing/sanitizers/coverage
  - being able to make statements about security
- triage/tracking of bug reports
  - helps stable releases, prevents lost bugs/fixes

"How"



Task Force / Funded

- testing infrastructure
- static analysis infrastructure
- testing frameworks
- fuzzing frameworks
- patch/bug tracking system
- ...

- evangelism
- bootstrapping
- setting an example
- monitoring
- taking care of special cases
- ...

Collective / Adopted

- adding tests/fuzzers
- keeping tests/fuzzers working
- resolving static warnings
- triaging/closing bugs
- marking fixes for stable
- ...

Thank you!

Linux is the model of technology innovation.
Let's make Linux **the model of OSS security**.

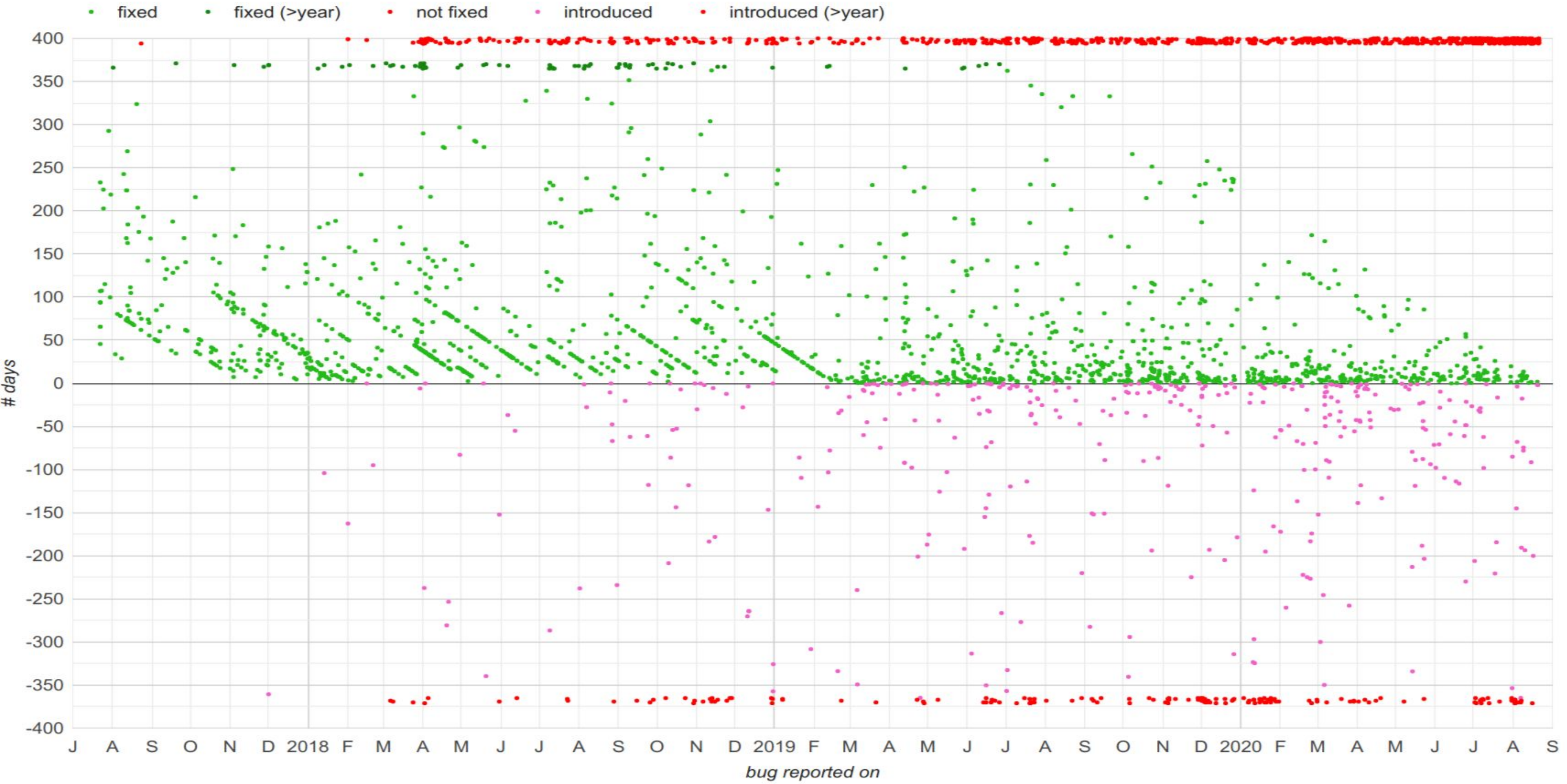
Dmitry Vyukov (dvyukov@)

Backup

Sample of stable backports

```
5b6717c6a3c0c USB: handle NULL config in usb_find_alt_setting()
4253abe6a3aac USB: fix error handling in usb_driver_claim_interface()
5eaaa5e9bd568 regulator: fix crash caused by null driver data
b6adc1f24bb35 spi: rspi: Fix interrupted DMA transfers
082e34f367a54 spi: rspi: Fix invalid SPI use during system suspend
6074b71d617dd spi: sh-msiof: Fix handling of write value for SISTR register
d120858fca5f6 spi: sh-msiof: Fix invalid SPI use during system suspend
429773341c34c spi: tegra20-slink: explicitly enable/disable clock
dc89d37f9098c intel_th: Fix device removal logic
247cc73cd8f5e serial: cpm_uart: return immediately from console poll
2b7ba104769b4 tty: serial: lpuart: avoid leaking struct tty_struct
4fe780c1baec2 x86/mm: Expand static page table for fixmap space
04bc4dd86d0f2 floppy: Do not copy a kernel pointer to user memoryin FDGETPRM ioctl
f88e50ea03000 ARM: dts: dra7: fix DCAN node addresses
99795ed0c62d9 iio: 104-quad-8: Fix off-by-one error in register selection
a82a772da7508 Input: xen-kbdfront - fix multi-touch XenStore node's locations
91e30cae8903a fs/lock: skip lock owner pid translation in case we are in init_pid_ns
0c4439c444160 EDAC: Fix memleak in module init error path
a4f7bea878871 nfsd: fix corrupted reply to badly ordered compound
de6ccdbd77345 gpio: Fix wrong rounding in gpio-menz127
5bcbbadf6ac54 module: exclude SHN_UNDEF symbols from kallsyms api
05f78b1a0e0c7 ASoC: dapm: Fix potential DAI widget pointer deref when linking DAIs
3fd534a5480ec EDAC, i7core: Fix memleaks and use-after-free on probe and remove
c96c2f2b11b6a scsi: megaraid_sas: Update controller info during resume
a56b97a2fc2d6 iomap: complete partial direct I/O writes synchronously
13ab355240a9d scsi: bnx2i: add error handling for ioremap_nocache
```

Bug Lifetimes


fixed
fixed (>year)
not fixed
introduced
introduced (>year)
# days
J
A
S
O
N
D
F
M
A
M
J
J
A
S
O
N
D
F
M
A
M
J
J
A
S
O
N
D
F
M
A
M
J
J
A
S
bug reported on