syzbot: automated kernel testing

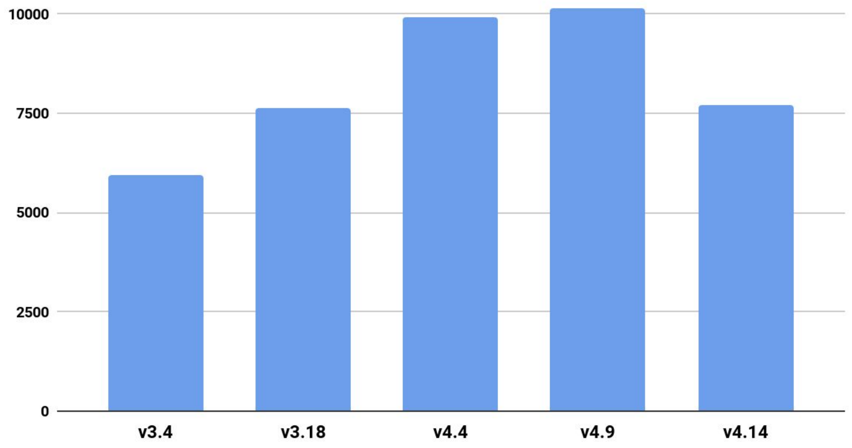
Linux Plumbers Conference 2018
Nov 13, 2018, Vancouver
Dmitry Vyukov, dvyukov@

Agenda

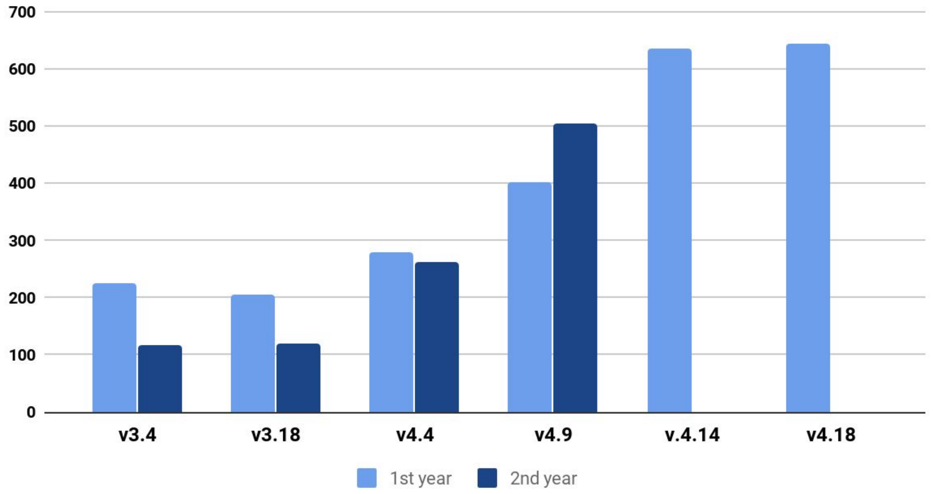
- Motivation
- syzbot workflow
- Pain points/wishes
- Future work

"Stable" releases

Number of backports in "stable" branches



Backports/month



"Stable" releases

- >95% of backports are fixes
- + not backported fixes (700+)
- + not fixed upstream bugs (300+)
- + not found bugs (XXXX+)
- + not detectable yet bugs (XXXX+) (info leaks, races)

Every "looks good and stable" release contains >**20'000 bugs**.

No, not getting better over time.

syzkaller/syzbot

syzkaller: kernel fuzzer

- grammar-based
- coverage-guided
- open-source: github.com/google/syzkaller

syzbot: automation on top of syzkaller

- continuous kernel/syzkaller build
- automatic reporting
- dashboard: syzkaller.appspot.com

syzbot report

```
SUBJECT: BUG: corrupted list in locks_delete_block
TO: linux-fsdevel@, linux-kernel@, jlayton@, viro@

HEAD commit:    442b8cea2477 Add linux-next specific files for 20181109
git tree:       linux-next
console output: https://syzkaller.appspot.com/x/log.txt?x=12b1262b400000
kernel config:  https://syzkaller.appspot.com/x/.config?x=2f72bdb11df9fbe8
dashboard link: https://syzkaller.appspot.com/bug?extid=13eb7470890c56ce3f37
C reproducer:   https://syzkaller.appspot.com/x/repro.c?x=11b5fa2b400000

------------[ cut here ]------------
kernel BUG at lib/list_debug.c:53!
Call Trace:
 __list_del_entry include/linux/list.h:117 [inline]
 locks_delete_block+0xce/0x3d0 fs/locks.c:716
 locks_mandatory_area+0x48b/0x6a0 fs/locks.c:1398
 rw_verify_area+0x2f2/0x360 fs/read_write.c:386
 vfs_writev+0x1f1/0x360 fs/read_write.c:1004
...
```

Bug stats

	Reported	Fixed	Fixed, %
Upstream (syzbot)	1400	960	69
Upstream (manual)	560	?	?
Internal (syzbot)	1740	388	22
Internal (manual)	470	?	?
Fuchsia	70	7	10
OpenBSD	20	10	50
gVisor	120	80	67
Akaros	35	3	9
Total	4415		

Manual bug reporting

Discover =>	Assess =>	Report =>	Ping =>	Support =>	Test =>	Fixed
[automated]	dup?	symbolize		answer questions		
	non-actionable?	find maintainers				
		find commit				
		find config				
		compose report				

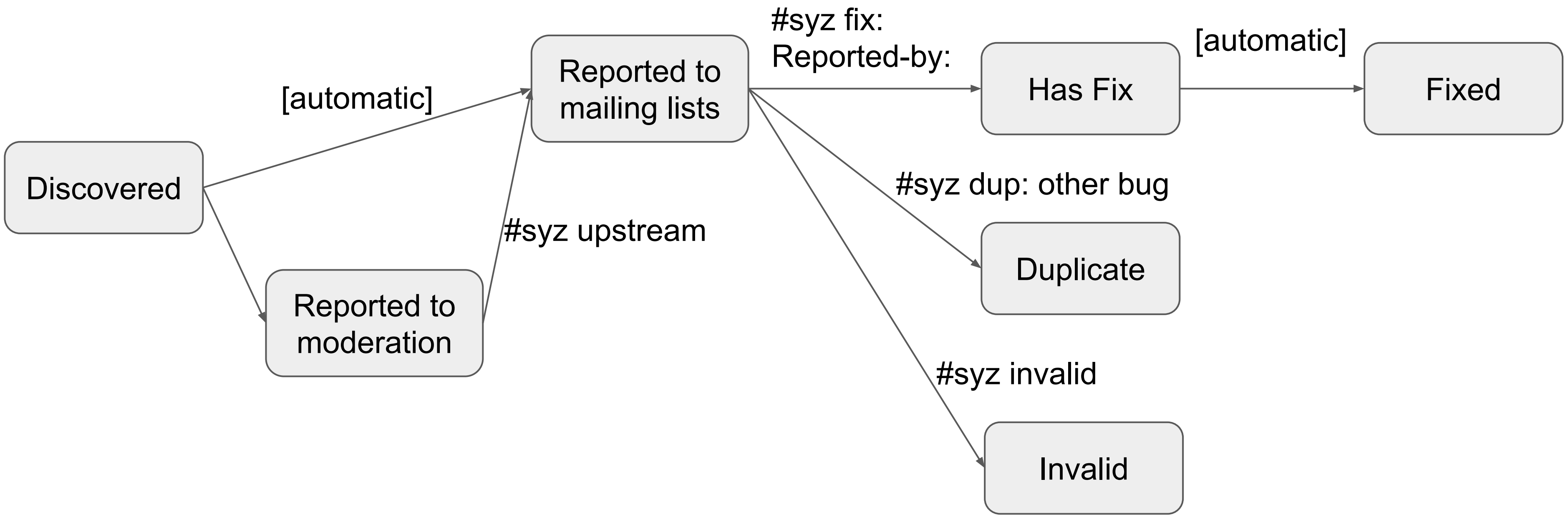
Works fine only you have reported one-two bugs.

Automated bug reporting [syzbot]

Discover => Report => Triage => Debug => Write test => Fix => Test => Mail => Fixed

[automated] [automated] [-------------- still on the developer --------------] [aided] [automated]

Bug life-cycle



Patch testing

For a bug with a reproducer, reply with:

```
#syz test: git://repo/address.git branch
#syz test: git://repo/address.git commit-hash
```
[optionally attach patch]

Can be used for:

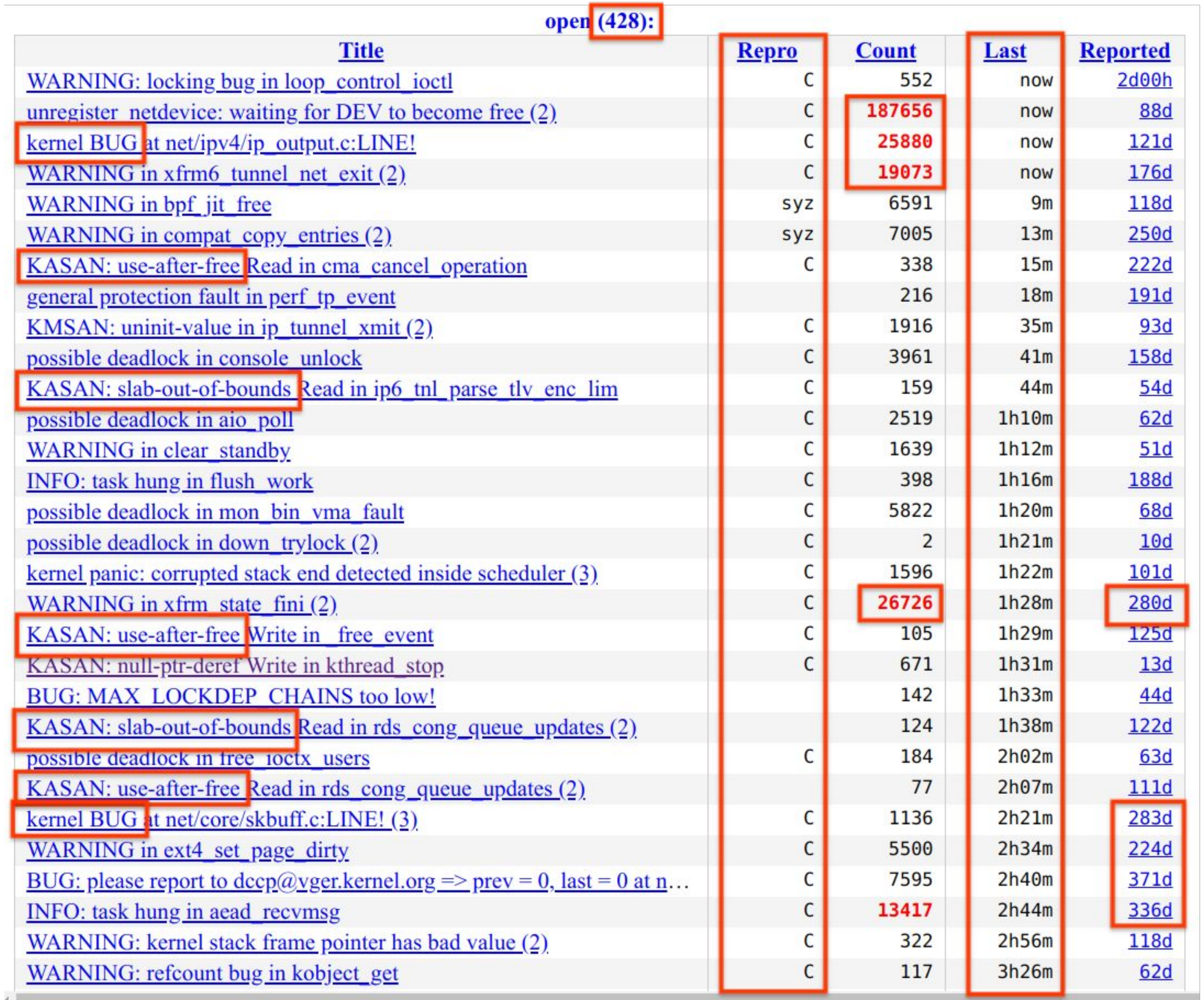
- fix patch testing
- retesting on latest HEAD (is it still happening?)
- observing other failure modes
- debugging (add additional checks, logging)

Reproducers

- Not all bugs have reproducers
  - Races/non-determinism
  - Accumulated state
  - Interactions between concurrent tests
  - ...
- Sometimes reproducers don't work for developers
  - The crash was triggered by the reproducer on fresh machine
  - Wrong source
  - Wrong config
  - No debugging configs
  - Different hardware
  - ...
- Fix ratio:
  - with repro: ~73%
  - w/o repro: 66%

Future automation

- bisection
- committed fix testing
- retesting on latest tree
- fix bisection
- pings
- auto-closing stale bugs


open (428):

Title	Repro	Count	Last	Reported
WARNING: locking bug in loop_control_ioctl	C	552	now	2d00h
unregister_netdevice: waiting for DEV to become free (2)	C	187656	now	88d
kernel BUG at net/ipv4/ip_output.c:LINE!	C	25880	now	121d
WARNING in xfrm6_tunnel_net_exit (2)	C	19073	now	176d
WARNING in bpf_jit_free	syz	6591	9m	118d
WARNING in compat_copy_entries (2)	syz	7005	13m	250d
KASAN: use-after-free Read in cma_cancel_operation	C	338	15m	222d
general protection fault in perf_tp_event		216	18m	191d
KMSAN: uninit-value in ip_tunnel_xmit (2)	C	1916	35m	93d
possible deadlock in console_unlock	C	3961	41m	158d
KASAN: slab-out-of-bounds Read in ip6_tnl_parse_tlv_enc_lim	C	159	44m	54d
possible deadlock in aio_poll	C	2519	1h10m	62d
WARNING in clear_standby	C	1639	1h12m	51d
INFO: task hung in flush_work	C	398	1h16m	188d
possible deadlock in mon_bin_vma_fault	C	5822	1h20m	68d
possible deadlock in down_trylock (2)	C	2	1h21m	10d
kernel panic: corrupted stack end detected inside scheduler (3)	C	1596	1h22m	101d
WARNING in xfrm_state_fini (2)	C	26726	1h28m	280d
KASAN: use-after-free Write in __free_event	C	105	1h29m	125d
KASAN: null-ptr-deref Write in kthread_stop	C	671	1h31m	13d
BUG: MAX_LOCKDEP_CHAINS too low!		142	1h33m	44d
KASAN: slab-out-of-bounds Read in rds_cong_queue_updates (2)		124	1h38m	122d
possible deadlock in free_ioctx_users	C	184	2h02m	63d
KASAN: use-after-free Read in rds_cong_queue_updates (2)		77	2h07m	111d
kernel BUG at net/core/skbuff.c:LINE! (3)	C	1136	2h21m	283d
WARNING in ext4_set_page_dirty	C	5500	2h34m	224d
BUG: please report to dccp@vger.kernel.org => prev = 0, last = 0 at n…	C	7595	2h40m	371d
INFO: task hung in aead_recvmsg	C	13417	2h44m	336d
WARNING: kernel stack frame pointer has bad value (2)	C	322	2h56m	118d
WARNING: refcount bug in kobject_get	C	117	3h26m	62d

Unfixed bugs

Hundreds of bugs are unfixed:

- Some are bad vulnerabilities
- Some are "just bugs"
- All harm syzkaller’s ability to uncover new bugs

Need help:

- Fixing
- Routing
- Duping
- Invalidating

Syscall Descriptions

syzkaller is based on declarative descriptions of system calls:

```
open(file filename, flags flags[open_flags],
     mode flags[open_mode]) fd
read(fd fd, buf buffer[out], size len[buf])
close(fd fd)
```

Tests **only** what's described.

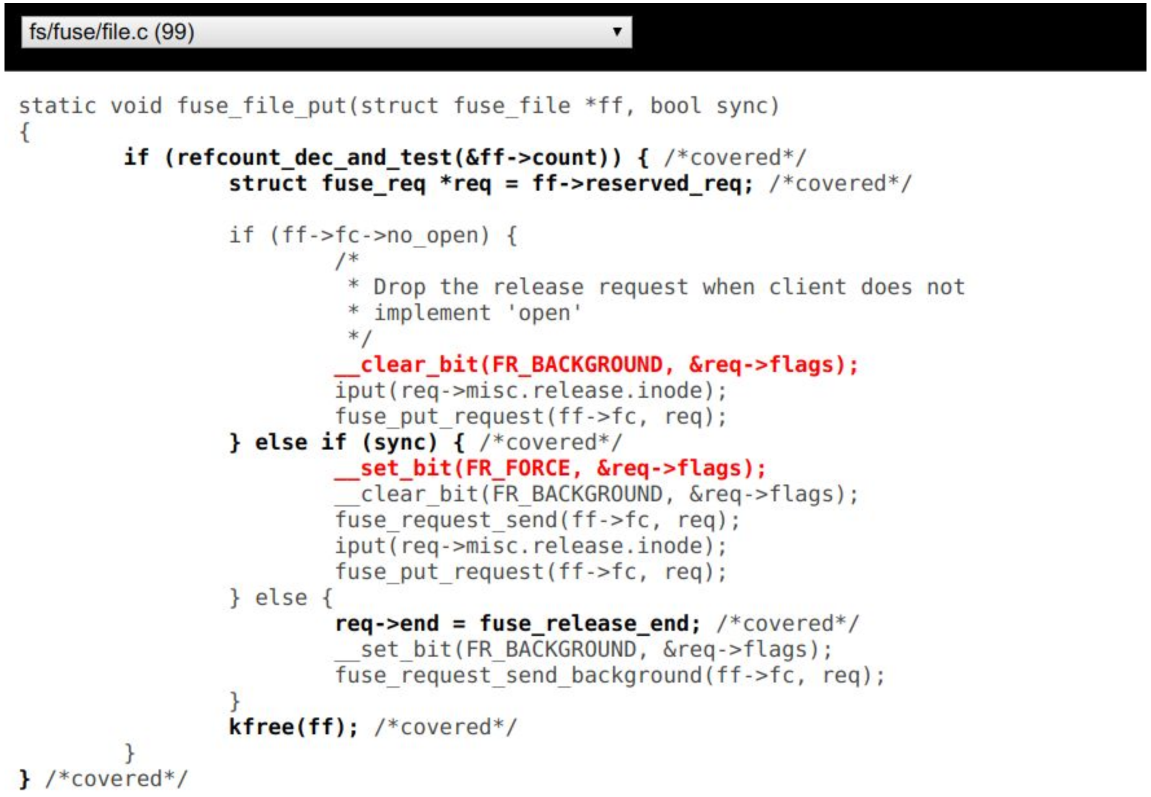
FUSE example

```
resource fd_fuse[fd]

open(file ptr[in, string["/dev/fuse"]],
        flags const[O_RDWR], mode const[0]) fd_fuse

write(fd fd_fuse, arg ptr[in, fuse_out[fuse_open_out]],
        len bytesize[arg])

fuse_open_out {
    fh          const[0, int64]
    open_flags flags[fuse_open_flags, int32]
    padding     const[0, int32]
}
```

Syscall descriptions

- Check if your subsystem has descriptions
- Check if necessary configs are enabled
- Check if it needs cmdline args, sysctls, setup
- Check how well it is tested
- Add descriptions

Coverage reports



```
static void fuse_file_put(struct fuse_file *ff, bool sync)
{
        if (refcount_dec_and_test(&ff->count)) { /*covered*/
                struct fuse_req *req = ff->reserved_req; /*covered*/

                if (ff->fc->no_open) {
                        /*
                         * Drop the release request when client does not
                         * implement 'open'
                         */
                        __clear_bit(FR_BACKGROUND, &req->flags);
                        iput(req->misc.release.inode);
                        fuse_put_request(ff->fc, req);
                } else if (sync) { /*covered*/
                        __set_bit(FR_FORCE, &req->flags);
                        __clear_bit(FR_BACKGROUND, &req->flags);
                        fuse_request_send(ff->fc, req);
                        iput(req->misc.release.inode);
                        fuse_put_request(ff->fc, req);
                } else {
                        req->end = fuse_release_end; /*covered*/
                        __set_bit(FR_BACKGROUND, &req->flags);
                        fuse_request_send_background(ff->fc, req);
                }
                kfree(ff); /*covered*/
        }
} /*covered*/
```

Stub/test devices

Examples:

- CONFIG_TUN (/dev/net/tun)
- CONFIG_VIDEO_VIVID (/dev/video0)
- CONFIG_MAC80211_HWSIM
- USB!

Allow to:

- write unit-tests for kernel (KernelCI, 0-day)
- test user-space code without hardware
- fuzz kernel in VMs

Need more of them!

Stub/test devices (contd)

Allow to reach:

- common code not reachable without a device
- external input paths
    - NFC
    - CAN
    - Bluetooth

Don'ts:

- single global device
- fixed number of devices
- only init_net namespace
- asynchronous processing

How you think kernel crashes look

```
WARNING: CPU: 0 PID: 4274 at drivers/dma-buf/dma-buf.c:992
CPU: 0 PID: 4274 Comm: syz-executor4 Not tainted 4.20.0-rc2
Hardware name: Google Compute Engine, BIOS Google 01/01/2011
Call Trace:
 vb2_vmalloc_detach_dmabuf+0x5a/0x80
 __vb2_plane_dmabuf_put.isra.5+0x122/0x310
 vb2_core_queue_release+0x62/0x80
 vb2_fop_release+0x77/0xc0
 vivid_fop_release+0x18e/0x440
 v4l2_release+0x224/0x3a0
 __fput+0x385/0xa30
 ____fput+0x15/0x20
 task_work_run+0x1e8/0x2a0
 exit_to_usermode_loop+0x318/0x380
 do_syscall_64+0x6be/0x820
```

How kernel crashes actually look

```
** 2158 printk messages dropped ** [   50.671305] Call Trace:
** 2378 printk messages dropped ** [   50.676929]  [<ffffffff81b0ce6d>] ? security_file_permission+0x13d/0x190
** 4635 printk messages dropped ** [   50.697826]  0000000000000000 3fe20028167234bc ffff8800b43179b0 ffffffff81cc9b0f
** 4555 printk messages dropped ** [   50.708497] Object ffff8801d3701170: 00 00 00 00 00 00 00 00 00 67 b4 b5 00 88 ff
ff  .........g......
** 5357 printk messages dropped ** [   50.721064]  ffff8801d3701080: fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc
** 4498 printk messages dropped ** [   50.731610]     __slab_alloc.isra.74.constprop.77+0x50/0xa0
** 3637 printk messages dropped ** [   50.740170]  ffff8801d3701280: fc fc fc fc fc fc fc fc fb fb fb fb fb fb fb fb
** 4491 printk messages dropped ** [   50.750742] INFO: Allocated in fasync_helper+0x29/0x90 age=1 cpu=1 pid=6024
** 4370 printk messages dropped ** [   50.761001]  [<ffffffff8123648d>] native_queued_spin_lock_slowpath+0x5ad/0x660
** 4510 printk messages dropped ** [   50.771609]                           ^
** 2979 printk messages dropped ** [   50.778606]     SyS_fcntl+0x5be/0xc70
** 3833 printk messages dropped ** [   50.794205]     run_ksoftirqd+0x20/0x60
** 4449 printk messages dropped ** [   50.811647]  [<ffffffff814d3af4>] print_trailer+0x114/0x1a0
** 3718 printk messages dropped ** [   50.820379]  0000000000000000 3fe20028167234bc ffff8800b43179b0 ffffffff81cc9b0f
** 4495 printk messages dropped ** [   50.830930]  [<ffffffff8123ab47>] do_raw_write_lock+0xc7/0x1d0
** 3497 printk messages dropped ** [   50.848107]  [<ffffffff81003044>] ? lockdep_sys_exit_thunk+0x12/0x14
** 4057 printk messages dropped ** [   50.857615]     run_ksoftirqd+0x20/0x60
** 3490 printk messages dropped ** [   50.872518]  [<ffffffff815bee10>] ? fsnotify+0xe40/0xe40
** 3600 printk messages dropped ** [   50.880974]     SyS_fcntl+0x5be/0xc70
** 4253 printk messages dropped ** [   50.906245]  [<ffffffff812cca9f>] ? do_futex+0xb2f/0x18a0
** 3636 printk messages dropped ** [   50.914820]  [<ffffffff814db1b7>] kasan_report.part.2+0x227/0x530
** 3921 printk messages dropped ** [   50.924057]     SyS_fcntl+0x5be/0xc70
** 2782 printk messages dropped ** [   50.930621]  [<ffffffff815bee10>] ? fsnotify+0xe40/0xe40
```

```
[  565.437862] WARNING: CPU: 0 PID: 19520 at ./arch/x86/include/asm/fpu/internal.h:340 __switch_to+0x10bd/0x13c0
[  565.455392] CPU: 0 PID: 19520 Comm: syz-executor6 Not tainted 4.15.0-rc6+ #246
[  565.472039] Call Trace:
[  566.523305] Shutting down cpus with NMI
[  566.527323] kasan: GPF could be caused by NULL-ptr deref or us[ e r5 6mem6.ory52 7a3cc30es]s
general protection fault: 0000 [#1] SMP KASAN
[[    55666.6.52572373383]8 ]      (f t(fratracece  bubufffeferr  eemmppttyy)
[[    556666..552277334488]]  CCPPUU::  11  PPIIDD::  3300558822  CComm: syz-executor3 Not tainted 4.15.0-rc6+ #246
[[    556666..552277336633]]  RRIIPP::  0000110:0:nanatitivev_e_wwrriittee__ccrr44++00xx4/40/x01x10
0

 Engine/Google Compute Engine, BIOS Google 01/01/2011
[[    556666..552277337700] ] RARXAX: : dffdffffcff00c0000000000000 00RB0X0:0  1fRfBXff: fd1f00ff0f0f0d7076090 007769
RCX: 0000000000000001
[[    556666..552277337766]]  RRBBPP::  ffffffffffee880000000033bbbb3300  RR0088::  ffffffffffee880000000033bbcc2288
RR0099::  ffffffffffee88000000003b3cbc686
8

[[    556666..552277338822] ] RR1133::  000000000000000000000000080028 2 R1R414: : ffffffff88808011bdbd666a6a66cc00
R R1515: : 11ffffffffffdd0000000000777777dd

[[    556666..552277338899]]  CCSS::    0001001 0D DS:S:  000000 00E ES:S : 0000000 0CR C0: R000:0
00000000008000005800003503
03
3
00) knlGS:0000000000000000
[[    556666..52572379369]6] Ca lClal lTr Tarcaec:e
:
0
2096f000 CR3: 00000001ced27006 CR4: 00000000001626e0
[[    556666..552277440088]]    ssmmp_p_ssttop_nmi_callback+0x45b/0x560
[[    556666..552727442266]]    ??  ppvvccllooocckk__rreeadad_f_fllaaggss++00xx116600//00xx116600
```

```
[   51.646683] =============================
[   51.650843] WARNING: suspicious RCU usage
[   51.655140] 4.15.0-rc6-mm1+ #52 Not tainted
[   51.659500] -----------------------------
[   51.663774] net/netfilter/ipset/ip_set_core.c:2057 suspicious
rcu_dereference_protected() usage!
[   51.672856]
[   51.672856] other info that might help us debug this:
[   51.672856]
[   51.681286]
[   51.681286] rcu_scheduler_active = 2, debug_locks = 1
[   51.688049] 3 locks held by kworker/u4:5/3913:
[   51.692668]  #0:  ((wq_comp
```

Kernel crashes

- Is there a crash at all?
- When it starts/ends?
- What's its "identity"?
- Intermixed/split lines

Crash parsing

- 14 top level rules
    - INFO:
    - Booting the kernel.
    - UBSAN:
    - unregister_netdevice: waiting for
    - kernel BUG
    - Kernel BUG
    - invalid opcode:
- 74 sub-rules
- 400+ hardcoded function/file names, pieces of output, etc
- 350+ tests

WARN_ON: please use only for bugs

KASAN (KernelAddressSANitizer)

- Detects:
  - use-after-free
  - out-of-bounds on heap/stack/globals
- detects bugs at the point of occurrence
- outputs informative reports
- based on compiler instrumentation (gcc4.9+ or clang)
- fast: ~~2x slowdown, ~~2x memory overhead
- upstream in 4.3 kernel
- easy to use (CONFIG_KASAN=y)

KASAN: future work

- Print global var names
- Print stack frame description
- Collect and print `call_rcu()` stacks
- Instrument bitops
- Instrument DMA transfers
- Instrument skb linear buffer (?)

KMSAN (KernelMemorySANitizer)

KMSAN detects uses of uninitialized values.

Working version on github.

So far found 110 bugs.

requires **clang**

KTSAN (KernelThreadSANitizer)

KTSAN detects data races.

Frozen prototype on github.

Say NO to "benign" data races

Thanks!

Q&A

syzkaller@googlegroups.com

Dmitry Vyukov, dvyukov@

Backup

Sample of release 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
```