syzkaller / sanitizers:

status update

Dmitry Vyukov (dvyukov@)
Linux Plumbers, Aug 26 2020

Agenda

- sanitizers
- syzkaller/syzbot
- stats and graphs

KCSAN: ConcurrencySanitizer

Detects **data races**.

Upstream since 5.8.

See Data-race detection in the Linux kernel talk.

KMSAN: MemorySantizer

Detects **uses of uninitialized memory**.

Not upstream (github.com/google/kmsan)

Rebased, deployed on syzbot.

GWP-ASAN -> KFENCE

Low-overhead sampling memory error detector (see LPC2019 talk).

RFC "soon".

contributed by:
Daniel Axtens

KASAN: AddressSantizer

```
CONFIG_KASAN_VMALLOC=y

CONFIG_VMAP_STACK=y
```

contributed by:
Walter Wu

KASAN: AddressSantizer

26e760c9a7c8 rcu: kasan: record and print call_rcu() call stack

contributed by:
Walter Wu

KASAN: AddressSantizer

26e760c9a7c8 rcu: kasan: record and print call_rcu() call stack

```
BUG: KASAN: use-after-free in afs_manage_cell
...
Freed by task 3903:
 kfree              mm/slab.c:3756
 rcu_do_batch       kernel/rcu/tree.c:2428
 rcu_core           kernel/rcu/tree.c:2656
 __do_softirq       kernel/softirq.c:298
```

contributed by:
Walter Wu

KASAN: AddressSantizer

26e760c9a7c8 rcu: kasan: record and print call_rcu() call stack

```
BUG: KASAN: use-after-free in afs_manage_cell
...
Freed by task 3903:
 kfree              mm/slab.c:3756
 rcu_do_batch       kernel/rcu/tree.c:2428
 rcu_core           kernel/rcu/tree.c:2656
 __do_softirq       kernel/softirq.c:298

Last call_rcu():
 call_rcu           kernel/rcu/tree.c:2968
 afs_manage_cell    fs/afs/cell.c:751
 process_one_work   kernel/workqueue.c:2269
 worker_thread      kernel/workqueue.c:2415
```

contributed by:
Jann Horn

KASAN: AddressSantizer

2f004eea0fc8 x86/kasan: Print original address on #GP

contributed by:
Jann Horn

KASAN: AddressSantizer

2f004eea0fc8 x86/kasan: Print original address on #GP

```
general protection fault for non-canonical address 0xdffffc0000000000

general protection fault for non-canonical address 0xdffffc000000000c

general protection fault for non-canonical address 0xdffffc00e0dffffe
```

contributed by:
Jann Horn

KASAN: AddressSantizer

2f004eea0fc8 x86/kasan: Print original address on #GP

```
general protection fault for non-canonical address 0xdffffc0000000000
KASAN: null-ptr-deref in range [0x0000000000000-0x0000000000007]
```

```
general protection fault for non-canonical address 0xdffffc000000000c
KASAN: null-ptr-deref in range [0x0000000000060-0x0000000000067]
```

```
general protection fault for non-canonical address 0xdffffc00e0dffffe
KASAN: user-memory-access in range [0x000706fffff0-0x000706fffff7]
```

contributed by:
Andrey Konovalov

KASAN+ARM MTE

[PATCH 00/35] kasan: add hardware tag-based mode for arm64

MTE (Memory Tagging Extensions)

syzkaller + syzbot

syzkaller + syzbot

syzkaller - OS kernel fuzzer:

- code-coverage-guided
- input-structure-aware
- focus on automation
- multi-OS

syzkaller + syzbot

syzkaller - OS kernel fuzzer:

- code-coverage-guided
- input-structure-aware
- focus on automation
- multi-OS

syzbot - syzkaller automation:

- continuous kernel/syzkaller update
- bug reporting / tracking
- web dashboard

syzkaller.appspot.com

New ARCHes

- amd64
- 386
- arm
- arm64
- ppc64le

New ARCHes

contributed by:
Alexander Egorenkov
Tobias Klauser
Jouni Hogander

- amd64
- 386
- arm
- arm64
- ppc64le
- **mips64le**
- **riscv64**
- **s390x**

New ARCHes

contributed by:
Alexander Egorenkov
Tobias Klauser
Jouni Hogander

- amd64
- 386
- arm
- arm64
- ppc64le
- **mips64le**
- **riscv64**
- **s390x**

not tested on syzbot

contributed by:
syzkaller project authors

Moar kernel interface descriptions

- watch_queue, copy_file_range, process_madvise, pidfd_getfd, preadv2
- /dev/snd/hw*, /dev/sequencer, /dev/raw
- wireguard
- exfat, afs
- v4l2: vim2m/vim2m2
- CAN/j1939
- netlabel/conntrack/ipset
- lwtunnel_encap
- NFNL_SUBSYS_(NFTABLES|CTNETLINK|ACCT|OCF|ULOG|QUEUE|CTHELPER)
- RTM_(NEW|DEL|GET)(MDB|VLAN|ADDRLABEL|LINKPROP)
- NETLINK_(RDMA|AUDIT|SOCK_DIAG)
- IPPROTO_(MPTCP|L2TP)
- AF_PHONET
- wirt_wifi, vlan, macvlan, ipvlan, xfrm, vlan, macvlan, ipvlan, mactap, geneve, macvtap, batadv
- more BPF
- ...

contributed by:
Andy Nguyen

BlueTooth

- inject external packets via /dev/vhci

contributed by:
Andy Nguyen

BlueTooth

- inject external packets via /dev/vhci
- pre-setup VHCI in test process
  - open(/dev/hvci)
  - socket(AF_BLUETOOTH, SOCK_RAW, BTPROTO_HCI)
  - ioctl's + read's + write's + thread's

contributed by:
Andy Nguyen

BlueTooth

- inject external packets via /dev/vhci
- pre-setup VHCI in test process
  - open(/dev/hvci)
  - socket(AF_BLUETOOTH, SOCK_RAW, BTPROTO_HCI)
  - ioctl's + read's + write's + thread's
- increases coverage of other BlueTooth parts

BlueTooth Bugs (<1 month)

```
KASAN: use-after-free Write in __sco_sock_close
KASAN: use-after-free Write in sco_chan_del
KASAN: use-after-free Read in hci_chan_del
KASAN: use-after-free Read in hci_send_acl
KASAN: use-after-free Read in __sco_sock_close
KASAN: use-after-free Read in hci_get_auth_info
KASAN: use-after-free Write in hci_conn_del
KASAN: use-after-free Read in __queue_work
KASAN: slab-out-of-bounds Read in lock_sock_nested
KASAN: slab-out-of-bounds Read in hci_inquiry_re...
KASAN: slab-out-of-bounds Read in hci_extended_i...
KASAN: slab-out-of-bounds Read in hci_le_meta_evt
KASAN: null-ptr-deref in amp_read_loc_assoc_final..
KASAN: null-ptr-deref Write in l2cap_chan_put
BUG: corrupted list in kobject_add_internal
BUG: unable to handle paging request in lock_sock_.
BUG: corrupted list in hci_chan_del
BUG: corrupted list in bt_accept_unlink
memory leak in hci_conn_add
memory leak in read_adv_mon_features
```

```
KMSAN: uninit-value in process_adv_report
KMSAN: uninit-value in hci_chan_lookup_handle
general protection fault in hci_phy_link_complete_.
general protection fault in hci_event_packet
general protection fault in bt_accept_unlink
WARNING: refcount bug in l2cap_global_chan_by_psm
WARNING: ODEBUG bug in put_device
WARNING in cancel_delayed_work
WARNING: ODEBUG bug in cancel_delayed_work
WARNING: ODEBUG bug in bt_host_release
WARNING: locking bug in hci_dev_reset
WARNING in hci_conn_timeout
WARNING: locking bug in l2cap_chan_del
WARNING: refcount bug in bt_accept_dequeue
INFO: register non-static key in l2cap_chan_del
INFO: register non-static key in l2cap_chan_close
INFO: register non-static key in skb_dequeue
INFO: register non-static key in skb_queue_purge
inconsistent lock state in sco_conn_del
inconsistent lock state in sco_sock_timeout
```

contributed by:
Necip Fazil Yildiran

io_uring

```
fd = io_uring_create(sq_size, cq_size, ...)
addr = mmap(..., [specific_size], ..., fd, ...)
addr[specific_offset1] = specific_value1
addr[specific_offset2] = specific_value3
addr[specific_offset3] = specific_value3
```

contributed by:
Necip Fazil Yildiran

io_uring

```
fd = io_uring_create(sq_size, cq_size, ...)
addr = mmap(..., [specific_size], ..., fd, ...)
addr[specific_offset1] = specific_value1
addr[specific_offset2] = specific_value3
addr[specific_offset3] = specific_value3
```

Pseudo-syscalls to the rescue!

```
# Submit sqe into the sq_ring
syz_io_uring_submit(
        ring_ptr ring_ptr,
        sqes_ptr sqes_ptr,
        sqe ptr[in, io_uring_sqe],
        sqes_index int32)
```

contributed by:
Necip Fazil Yildiran

io_uring

KASAN: use-after-free Read in io_uring_setup (2)
KASAN: use-after-free Read in io_async_task_func
BUG: NULL pointer dereference in loop_rw_iter
general protection fault in io_poll_double_wake
possible deadlock in io_timeout_fn
possible deadlock in __io_queue_deferred
possible deadlock in io_queue_linked_timeout
KCSAN: data-race in __io_cqring_fill_event/io_uring_poll
INFO: task can't die in io_uring_flush
memory leak in io_submit_sqes
memory leak in rw_copy_check_uvector

contributed by:
Necip Fazil Yildiran

io_uring

KASAN: use-after-free Read in io_uring_setup (2)
KASAN: use-after-free Read in io_async_task_func
BUG: NULL pointer dereference in loop_rw_iter
general protection fault in io_poll_double_wake
possible deadlock in io_timeout_fn
possible deadlock in __io_queue_deferred
possible deadlock in io_queue_linked_timeout
KCSAN: data-race in __io_cqring_fill_event/io_uring_poll
INFO: task can't die in io_uring_flush
memory leak in io_submit_sqes
memory leak in rw_copy_check_uvector

BUG: kernel NULL pointer dereference
Call Trace:
 loop_rw_iter fs/io_uring.c:2829
 io_write+0x6a2/0x7a0 fs/io_uring.c:3190
 io_issue_sqe+0x1b0/0x60d0 fs/io_uring.c:5530
 io_wq_submit_work+0x183/0x3d0 fs/io_uring.c:5775
 io_worker_handle_work+0xa45/0x13f0 fs/io-wq.c:527
 io_wqe_worker+0xbf0/0x10e0 fs/io-wq.c:569
 kthread+0x3b5/0x4a0 kernel/kthread.c:292

contributed by:
Zubin Mithra

Fix Bisection

If we suspect a bug is fixed -> find what fixed it!

contributed by:
Zubin Mithra

Fix Bisection

If we suspect a bug is fixed -> find what fixed it!

```
Subject: Re: WARNING: locking bug in try_to_grab_pending
Date: Fri, 14 Aug 2020 06:17:07 -0700
Message-ID: <000000000000dbe6ee05acd63ca2@google.com> (raw)
In-Reply-To: <0000000000006dc0290581ca413e@google.com>

syzbot suspects this issue was fixed by commit:

commit 1378817486d6860f6a927f573491afe65287abf1
Author: Eric Dumazet <edumazet@google.com>
Date:   Thu May 21 18:29:58 2020 +0000

    tipc: block BH before using dst_cache

bisection log:  https://syzkaller.appspot.com/x/bisect.txt?x=175599f6900000
start commit:   6663cf82 flow_offload: Fix flow action infrastructure
git tree:       net-next
kernel config:  https://syzkaller.appspot.com/x/.config?x=8572a6e4661225f4
dashboard link: https://syzkaller.appspot.com/bug?extid=2b713236b28823cd4dff
syz repro:      https://syzkaller.appspot.com/x/repro.syz?x=13e932a8c00000

If the result looks correct, please mark the issue as fixed by replying with:

#syz fix: tipc: block BH before using dst_cache
```

contributed by:
Zubin Mithra

Fix Bisection

If we suspect a bug is fixed -> find what fixed it!

```
Subject: Re: WARNING: locking bug in try_to_grab_pending
Date: Fri, 14 Aug 2020 06:17:07 -0700
Message-ID: <000000000000dbe6ee05acd63ca2@google.com> (raw)
In-Reply-To: <0000000000006dc0290581ca413e@google.com>

syzbot suspects this issue was fixed by commit:

commit 1378817486d6860f6a927f573491afe65287abf1
Author: Eric Dumazet <edumazet@google.com>
Date:   Thu May 21 18:29:58 2020 +0000

    tipc: block BH before using dst_cache

bisection log:  https://syzkaller.appspot.com/x/bisect.txt?x=175599f6900000
start commit:   6663cf82 flow_offload: Fix flow action infrastructure
git tree:       net-next
kernel config:  https://syzkaller.appspot.com/x/.config?x=8572a6e4661225f4
dashboard link: https://syzkaller.appspot.com/bug?extid=2b713236b28823cd4dff
syz repro:      https://syzkaller.appspot.com/x/repro.syz?x=13e932a8c00000

If the result looks correct, please mark the issue as fixed by replying with:

#syz fix: tipc: block BH before using dst_cache
```

contributed by:
Zubin Mithra

Fix Bisection

If we suspect a bug is fixed -> find what fixed it!

```
Subject: Re: WARNING: locking bug in try_to_grab_pending
Date: Fri, 14 Aug 2020 06:17:07 -0700
Message-ID: <000000000000dbe6ee05acd63ca2@google.com> (raw)
In-Reply-To: <0000000000006dc0290581ca413e@google.com>

syzbot suspects this issue was fixed by commit:

commit 1378817486d6860f6a927f573491afe65287abf1
Author: Eric Dumazet <edumazet@google.com>
Date:   Thu May 21 18:29:58 2020 +0000

    tipc: block BH before using dst_cache

bisection log:  https://syzkaller.appspot.com/x/bisect.txt?x=175599f6900000
start commit:   6663cf82 flow_offload: Fix flow action infrastructure
git tree:       net-next
kernel config:  https://syzkaller.appspot.com/x/.config?x=8572a6e4661225f4
dashboard link: https://syzkaller.appspot.com/bug?extid=2b713236b28823cd4dff
syz repro:      https://syzkaller.appspot.com/x/repro.syz?x=13e932a8c00000

If the result looks correct, please mark the issue as fixed by replying with:

#syz fix: tipc: block BH before using dst_cache
```

contributed by:
Zubin Mithra

Fix Bisection

If we suspect a bug is fixed -> find what fixed it!

```
Subject: Re: WARNING: locking bug in try_to_grab_pending
Date: Fri, 14 Aug 2020 06:17:07 -0700
Message-ID: <000000000000dbe6ee05acd63ca2@google.com> (raw)
In-Reply-To: <0000000000006dc0290581ca413e@google.com>

syzbot suspects this issue was fixed by commit:

commit 1378817486d6860f6a927f573491afe65287abf1
Author: Eric Dumazet <edumazet@google.com>
Date:   Thu May 21 18:29:58 2020 +0000

    tipc: block BH before using dst_cache

bisection log:  https://syzkaller.appspot.com/x/bisect.txt?x=175599f6900000
start commit:   6663cf82 flow_offload: Fix flow action infrastructure
git tree:       net-next
kernel config:  https://syzkaller.appspot.com/x/.config?x=8572a6e4661225f4
dashboard link: https://syzkaller.appspot.com/bug?extid=2b713236b28823cd4dff
syz repro:      https://syzkaller.appspot.com/x/repro.syz?x=13e932a8c00000

If the result looks correct, please mark the issue as fixed by replying with:

#syz fix: tipc: block BH before using dst_cache
```

mailed 80 results

Config Bisection

contributed by:
Jukka Kaartinen
Jouni Hogander

Full syzbot .config
[lots of subsystems enabled]

Config Bisection

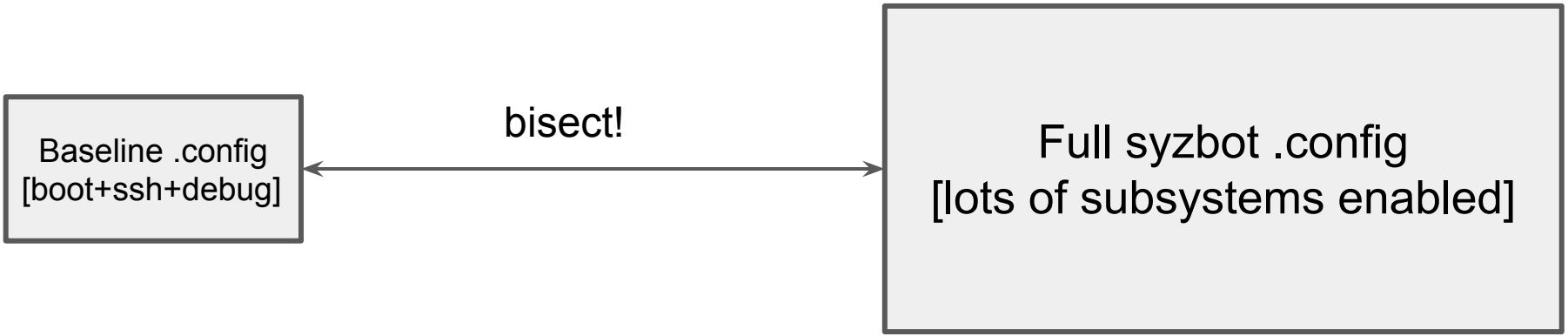
contributed by:
Jukka Kaartinen
Jouni Hogander

Baseline .config
[boot+ssh+debug]

Full syzbot .config
[lots of subsystems enabled]

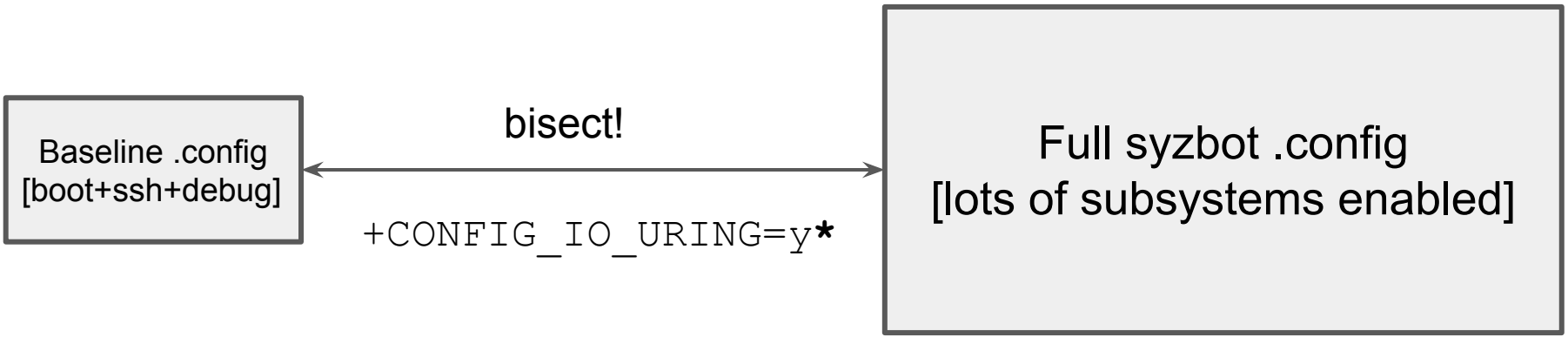
Config Bisection

contributed by:
Jukka Kaartinen
Jouni Hogander



contributed by:
Jukka Kaartinen
Jouni Hogander

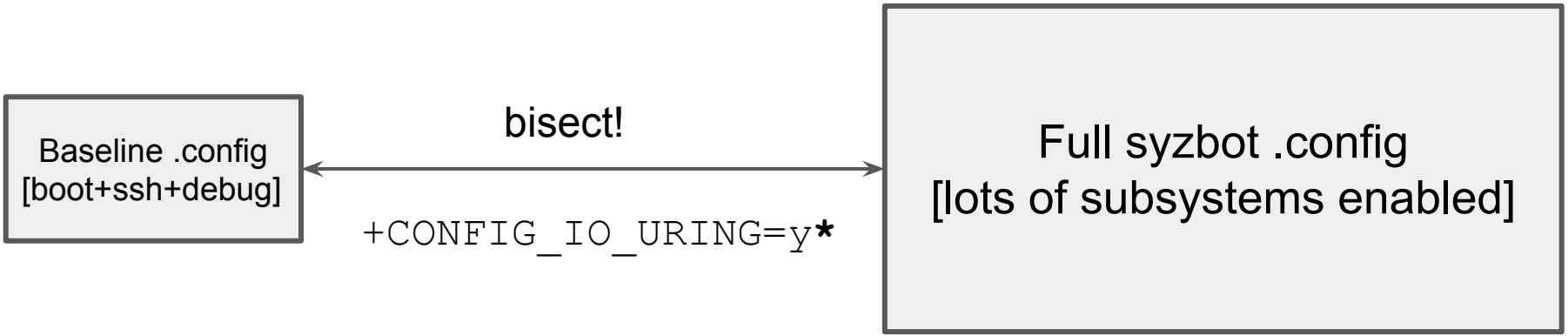
Config Bisection



*ANY REFERENCE TO REAL CONFIGS IS PURELY COINCIDENTAL

contributed by:
Jukka Kaartinen
Jouni Hogander

Config Bisection

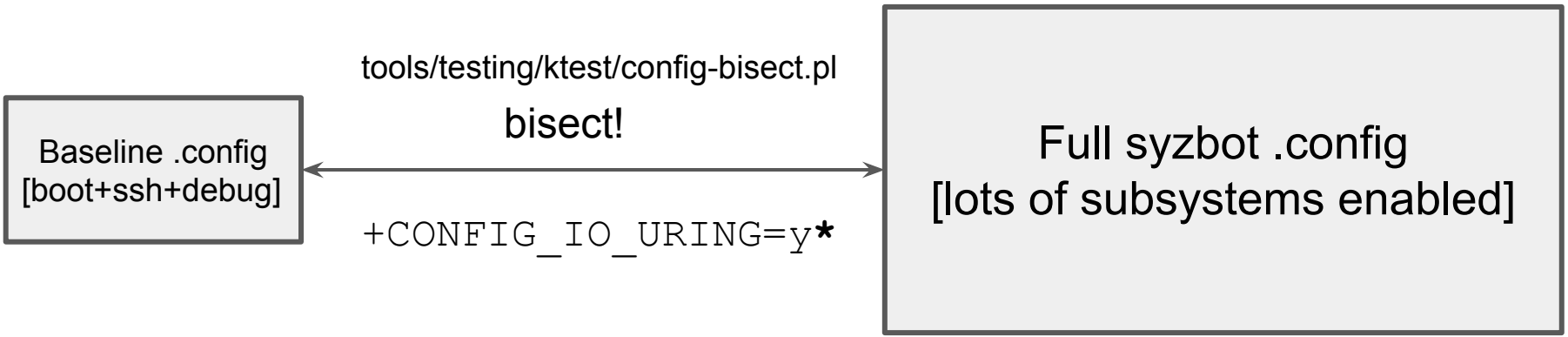


+ faster builds
+ fewer unrelated crashes
+ fewer broken builds/boots
+ smaller config in reproducers

*ANY REFERENCE TO REAL CONFIGS IS PURELY COINCIDENTAL

contributed by:
Jukka Kaartinen
Jouni Hogander

Config Bisection



+ faster builds
+ fewer unrelated crashes
+ fewer broken builds/boots
+ smaller config in reproducers

*ANY REFERENCE TO REAL CONFIGS IS PURELY COINCIDENTAL

More Automation

- Remind about unexisting fixing commit
  - can't be discovered anywhere for 90 days

More Automation

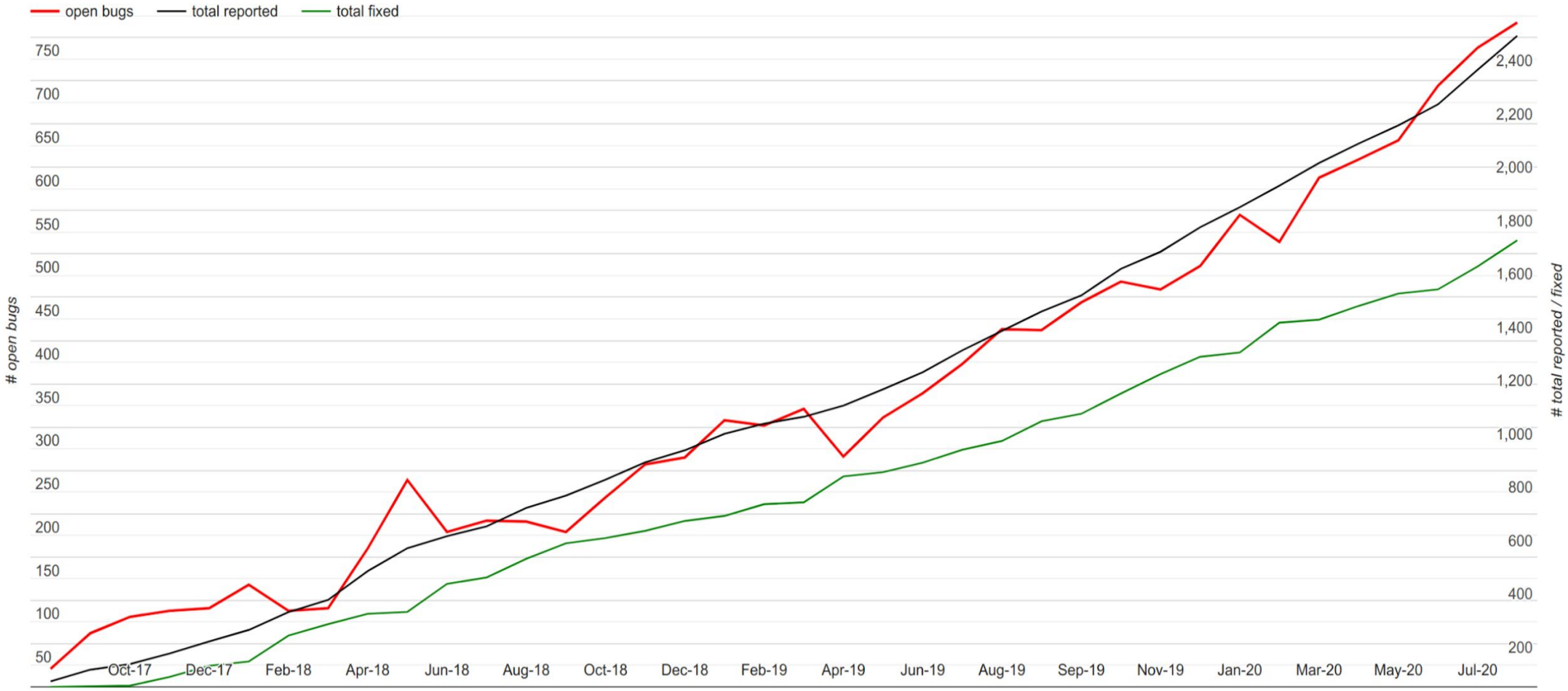
- Remind about unexisting fixing commit
  - can't be discovered anywhere for 90 days
- Auto-obsoleting
  - no reproducer
  - upstream: 80-120 days; linux-next: 40-60 days
  - based on crash rate

More Automation

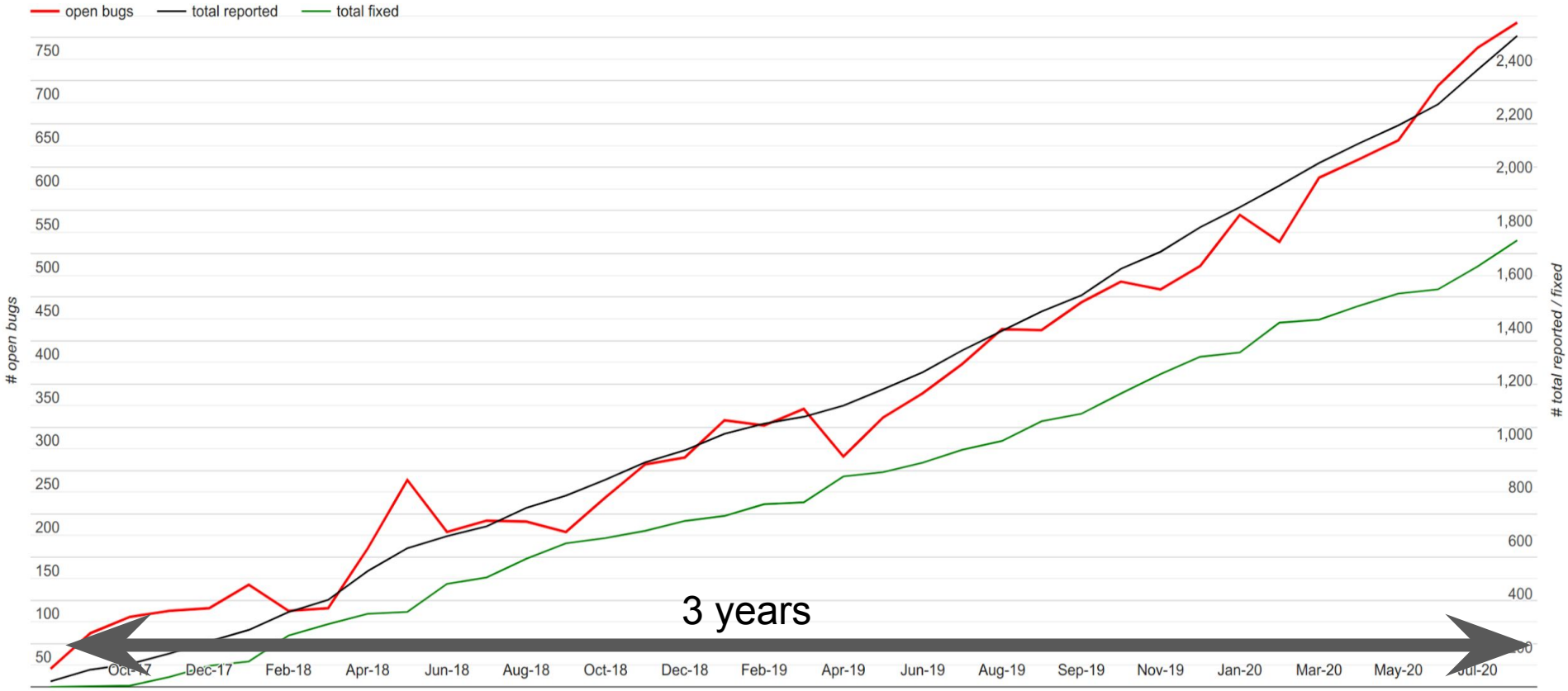
- Remind about unexisting fixing commit
  - can't be discovered anywhere for 90 days
- Auto-obsoleting
  - no reproducer
  - upstream: 80-120 days; linux-next: 40-60 days
  - based on crash rate
- Auto-upstreaming
  - syzkaller-upstream-moderation@googlegroups.com -> linux-kernel@vger.kernel.org
  - based on:
    - reproducer availability
    - number of crashes
    - bug type
    - tool (KCSAN/KMSAN)
  - edge-triggered -> level-triggered

stats and graphs

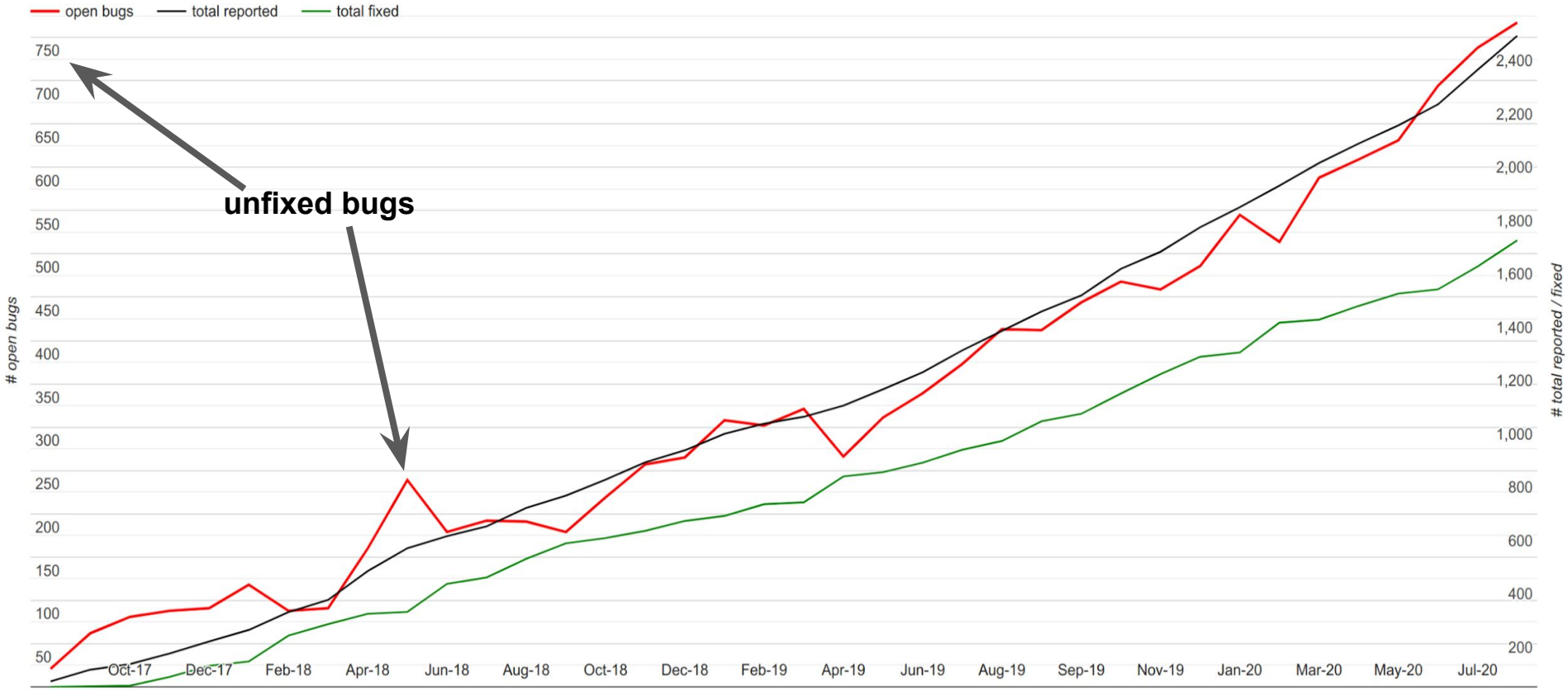
Upstream Bug Stats



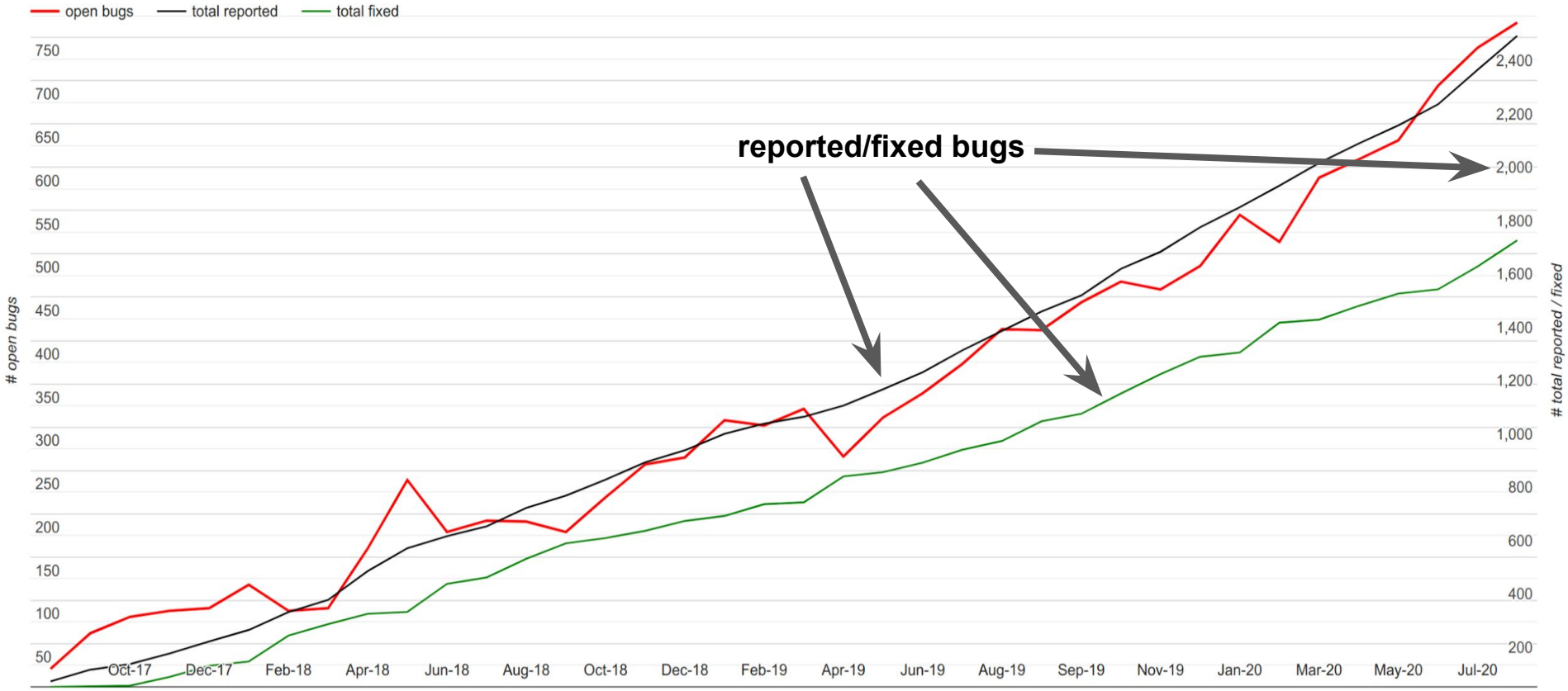
Upstream Bug Stats



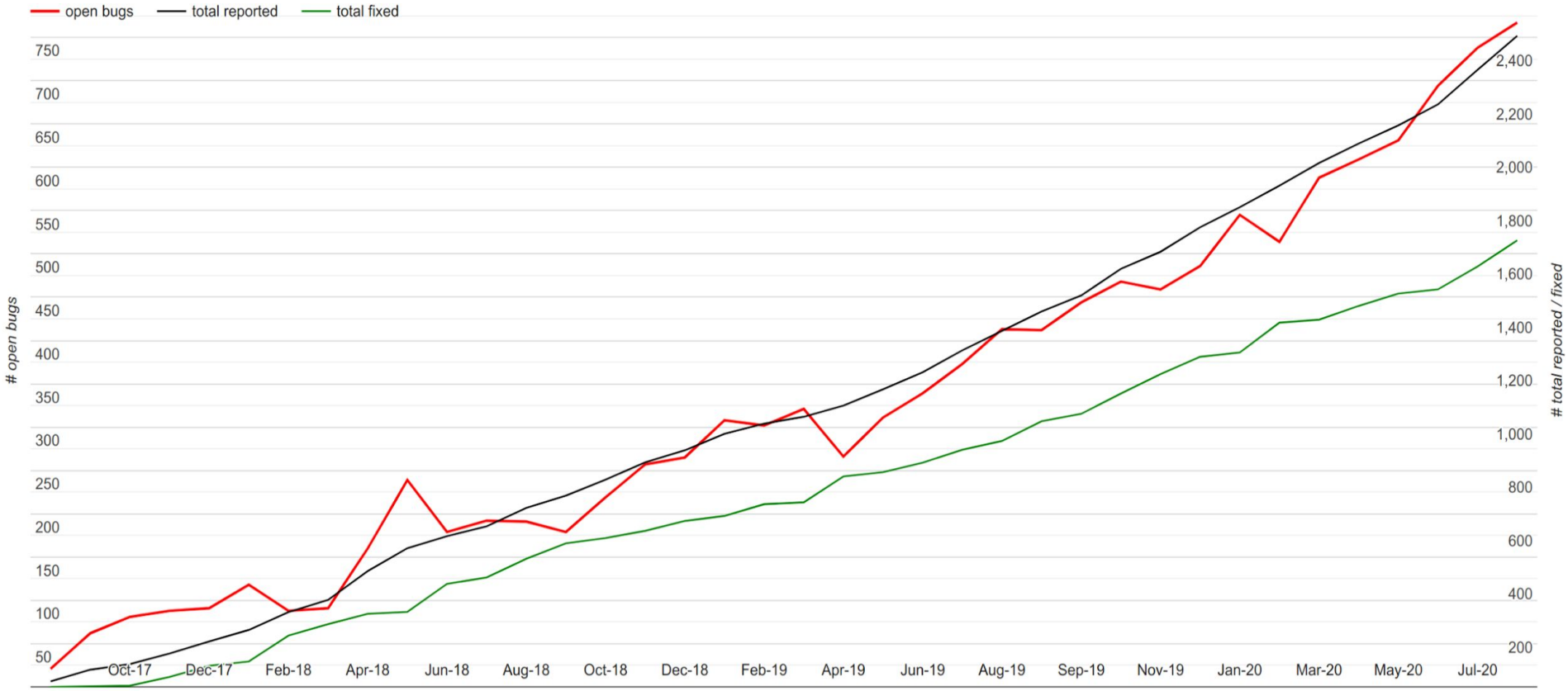
Upstream Bug Stats



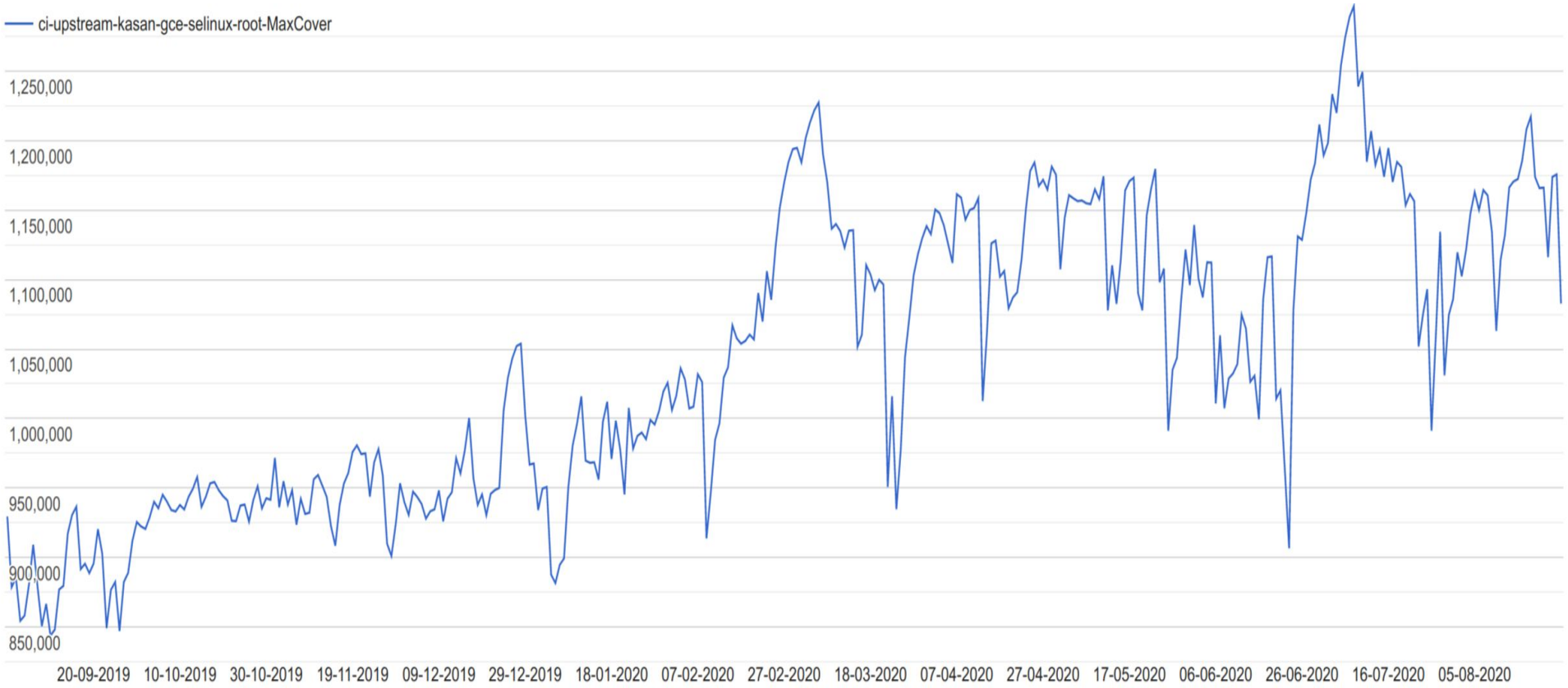
Upstream Bug Stats



Upstream Bug Stats



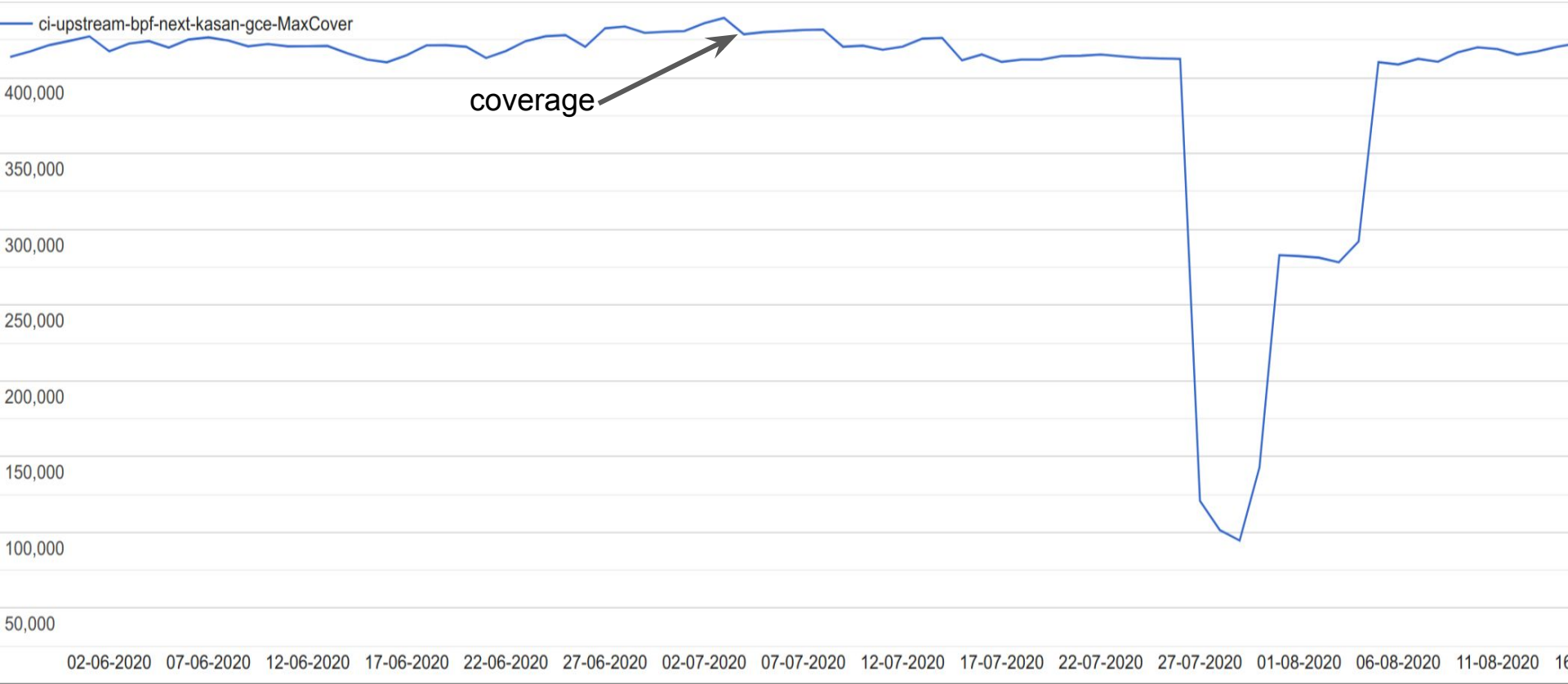
Upstream kernel coverage



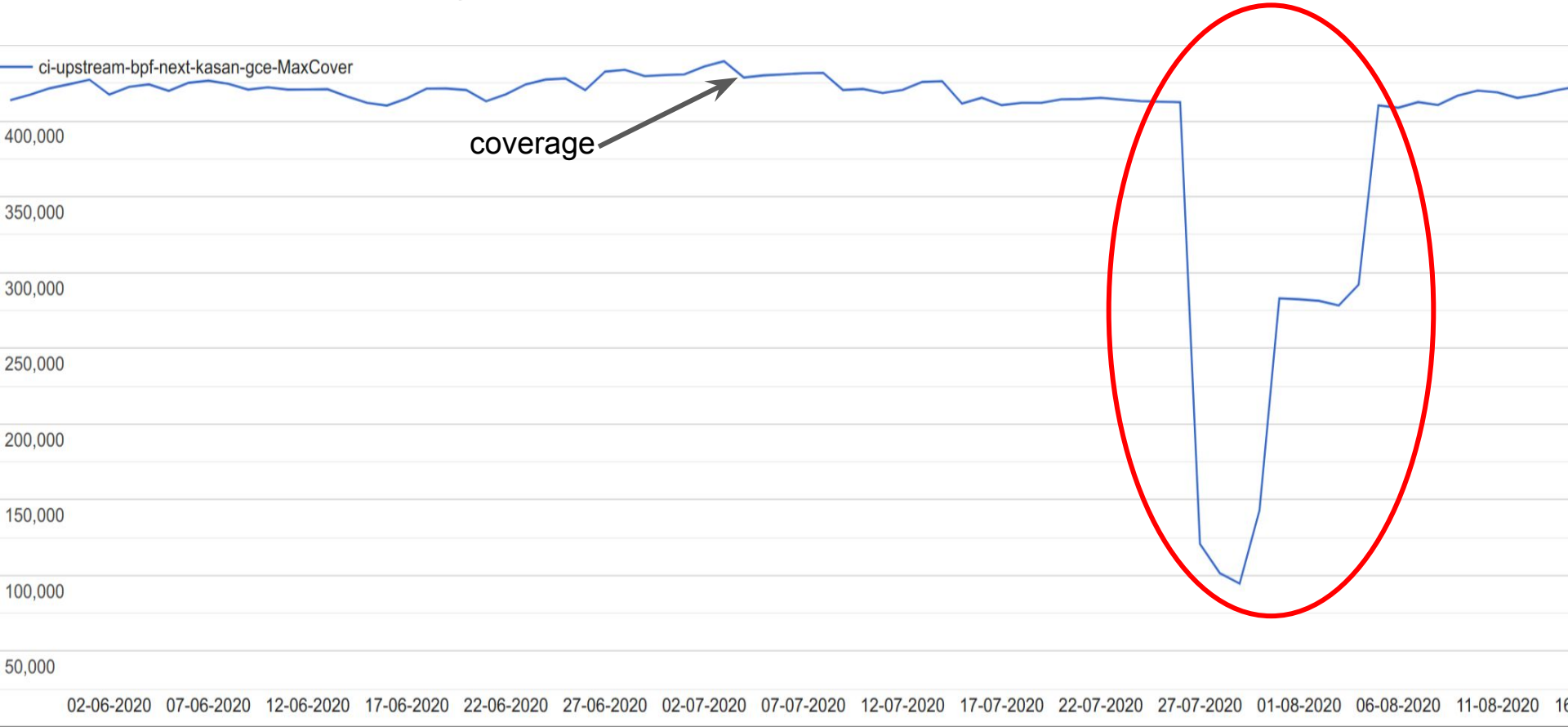
Upstream kernel coverage

Upstream kernel coverage

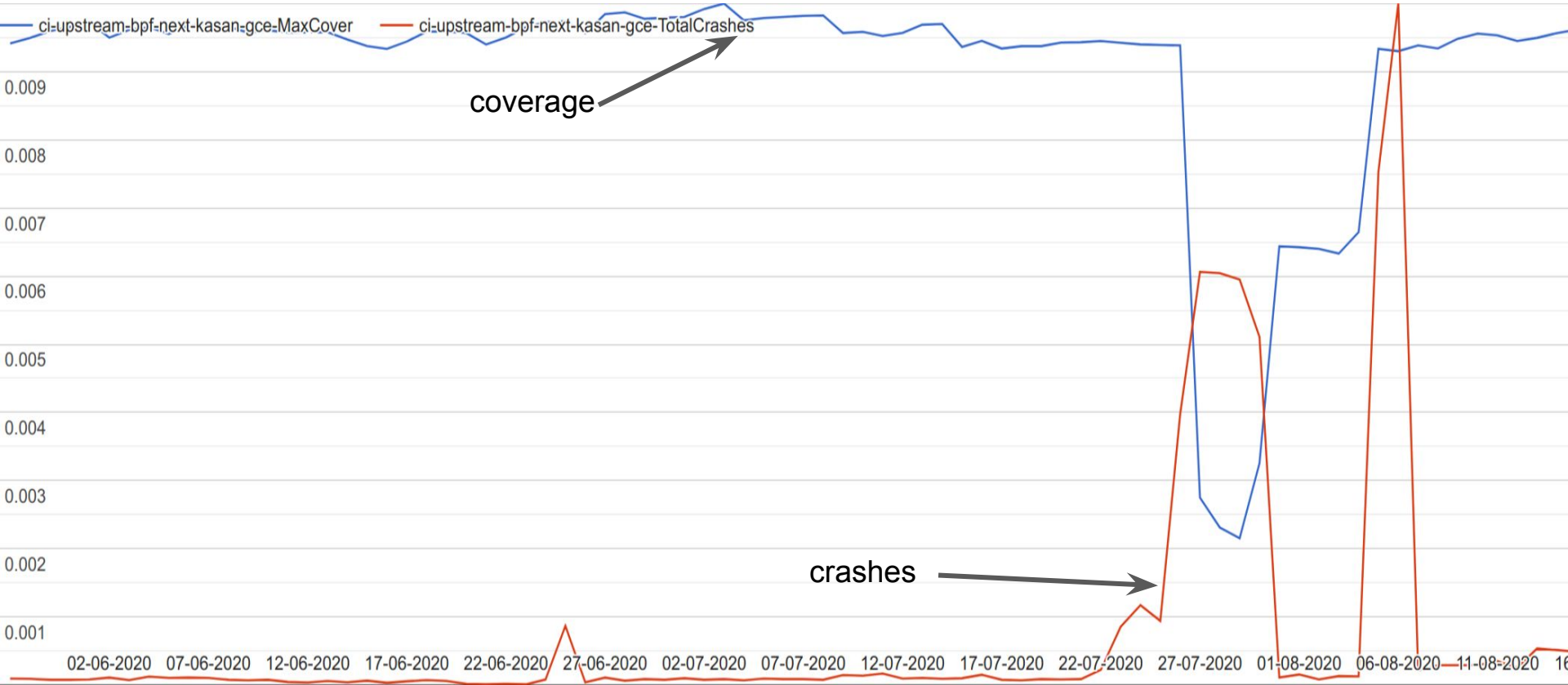
bpf-next coverage



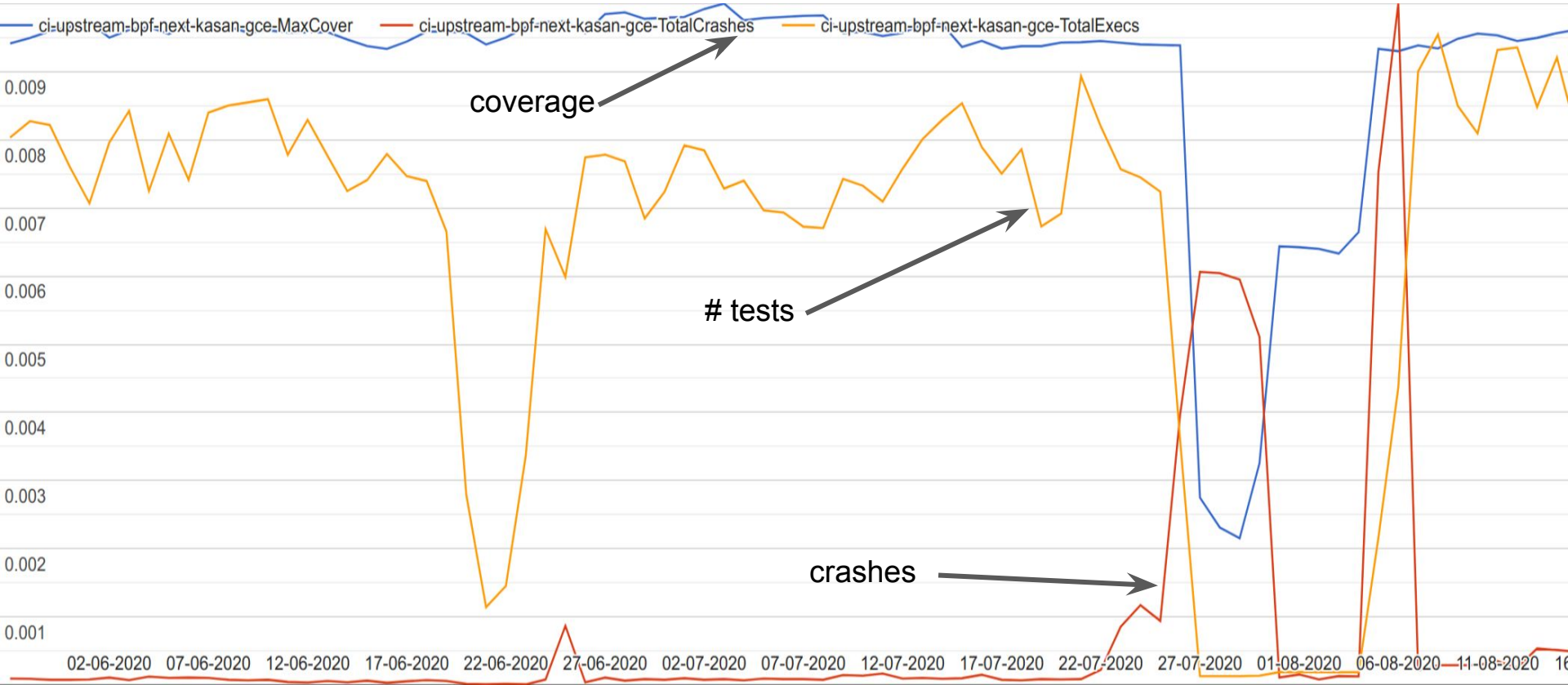
bpf-next coverage



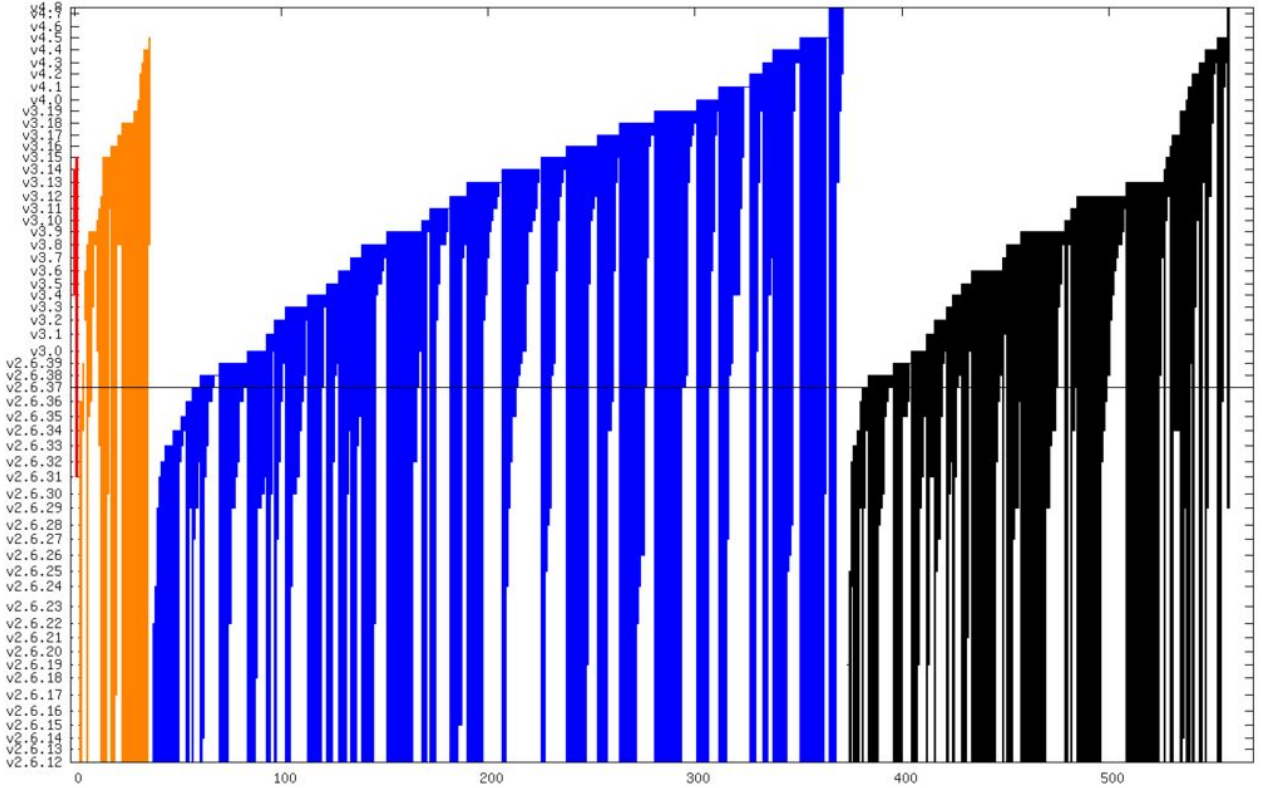
bpf-next coverage



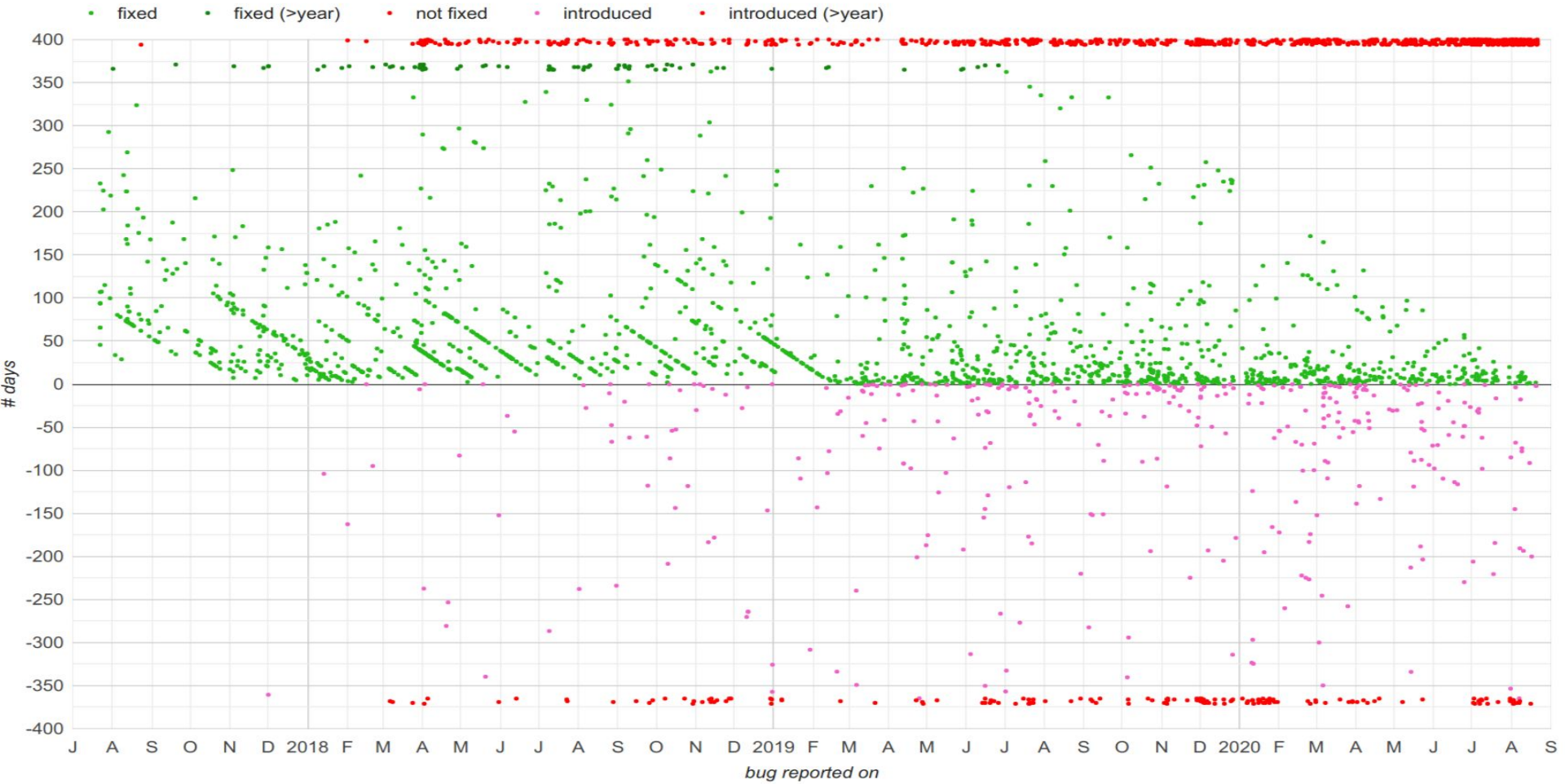
bpf-next coverage



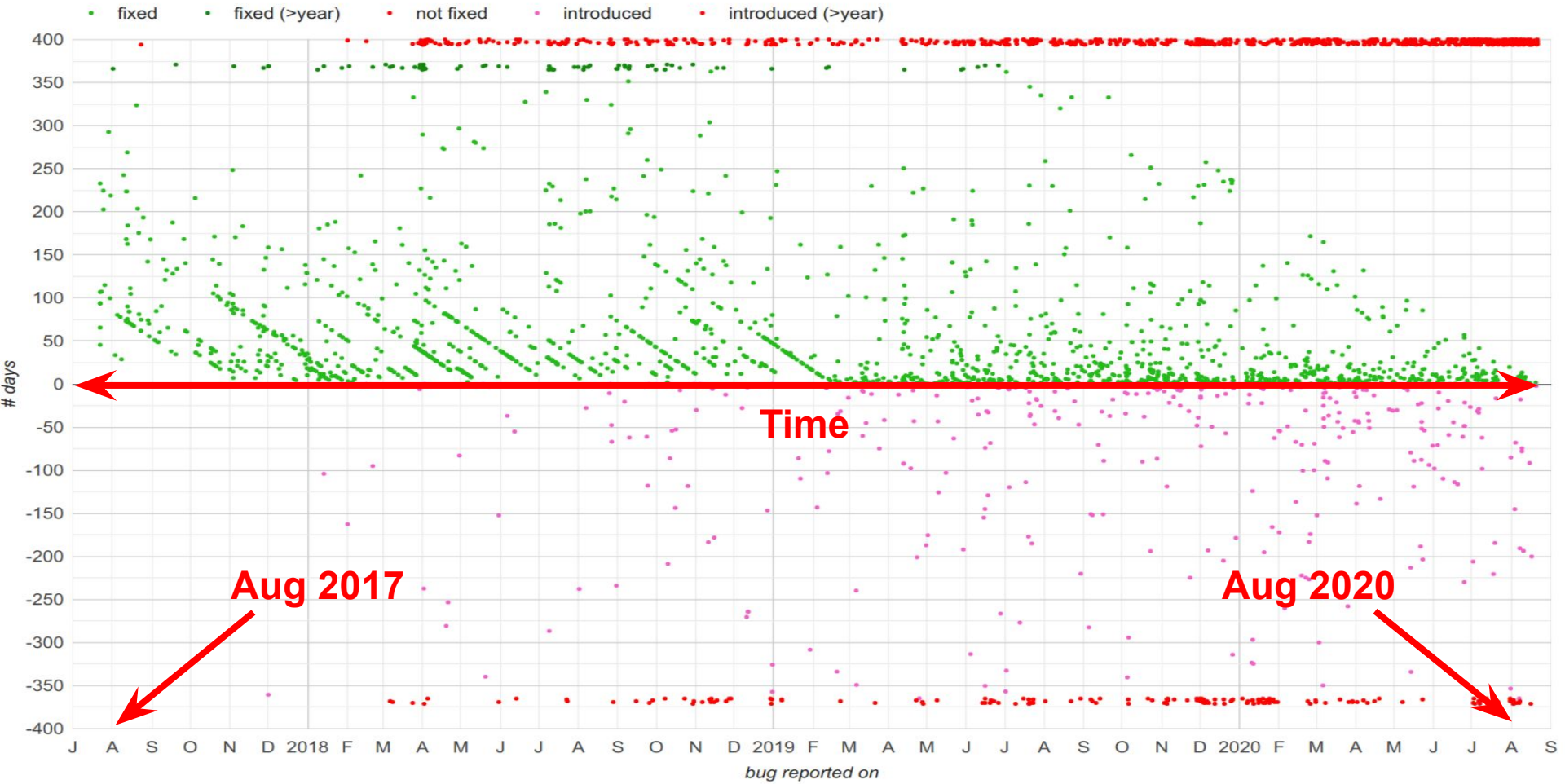
Security bug lifetime (© Kees Cook)



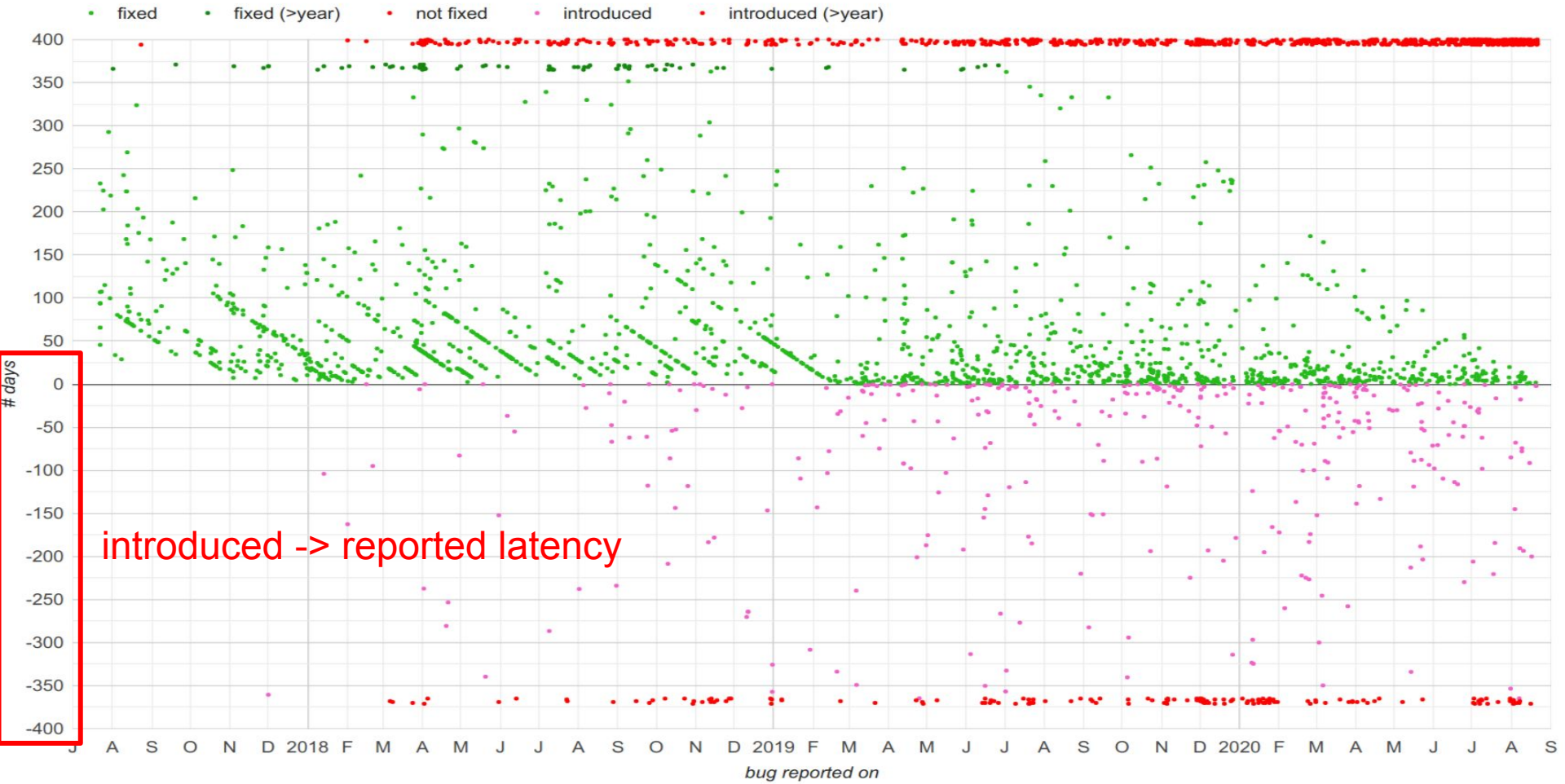
Bug Lifetimes



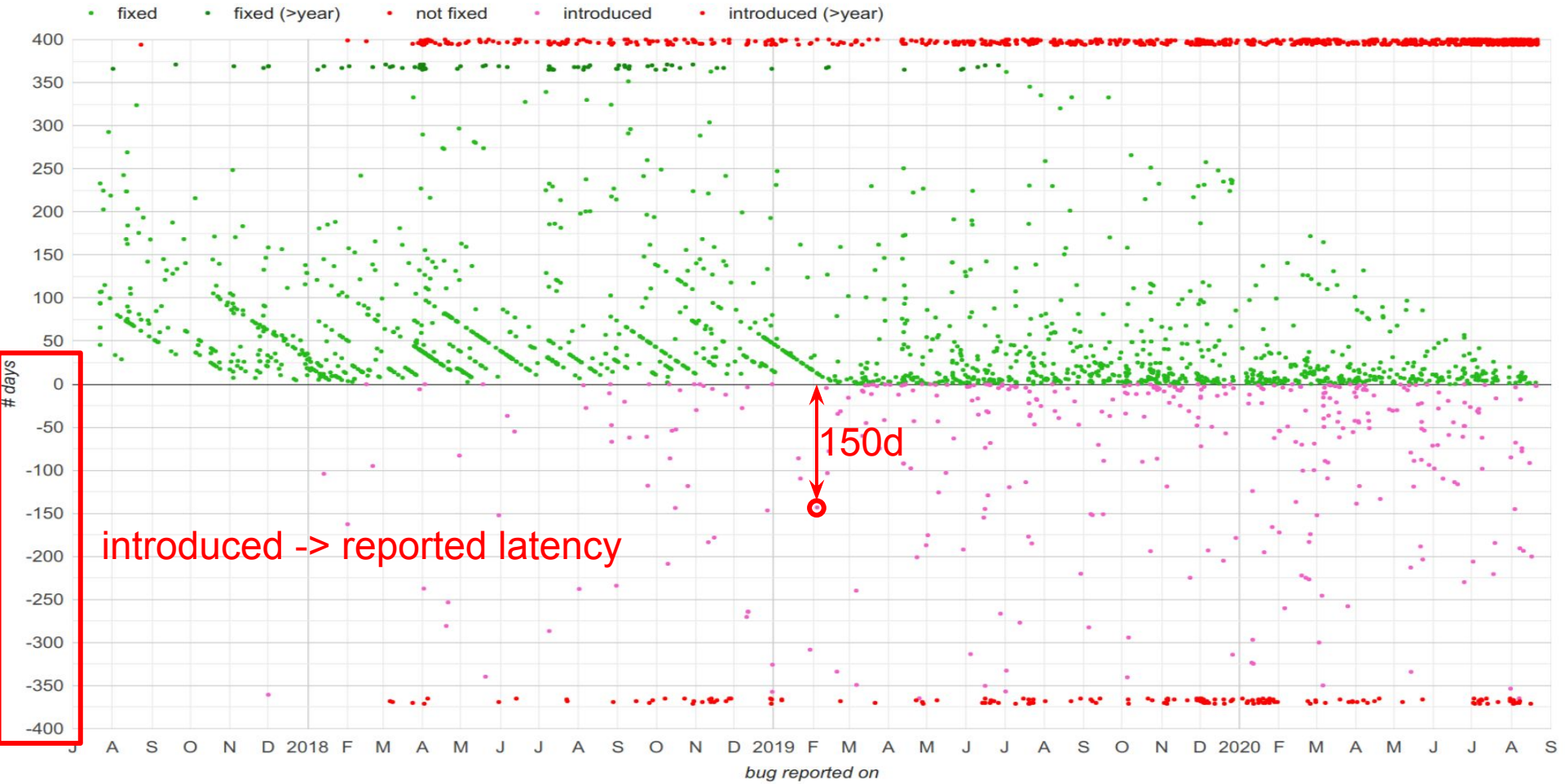
Bug Lifetimes



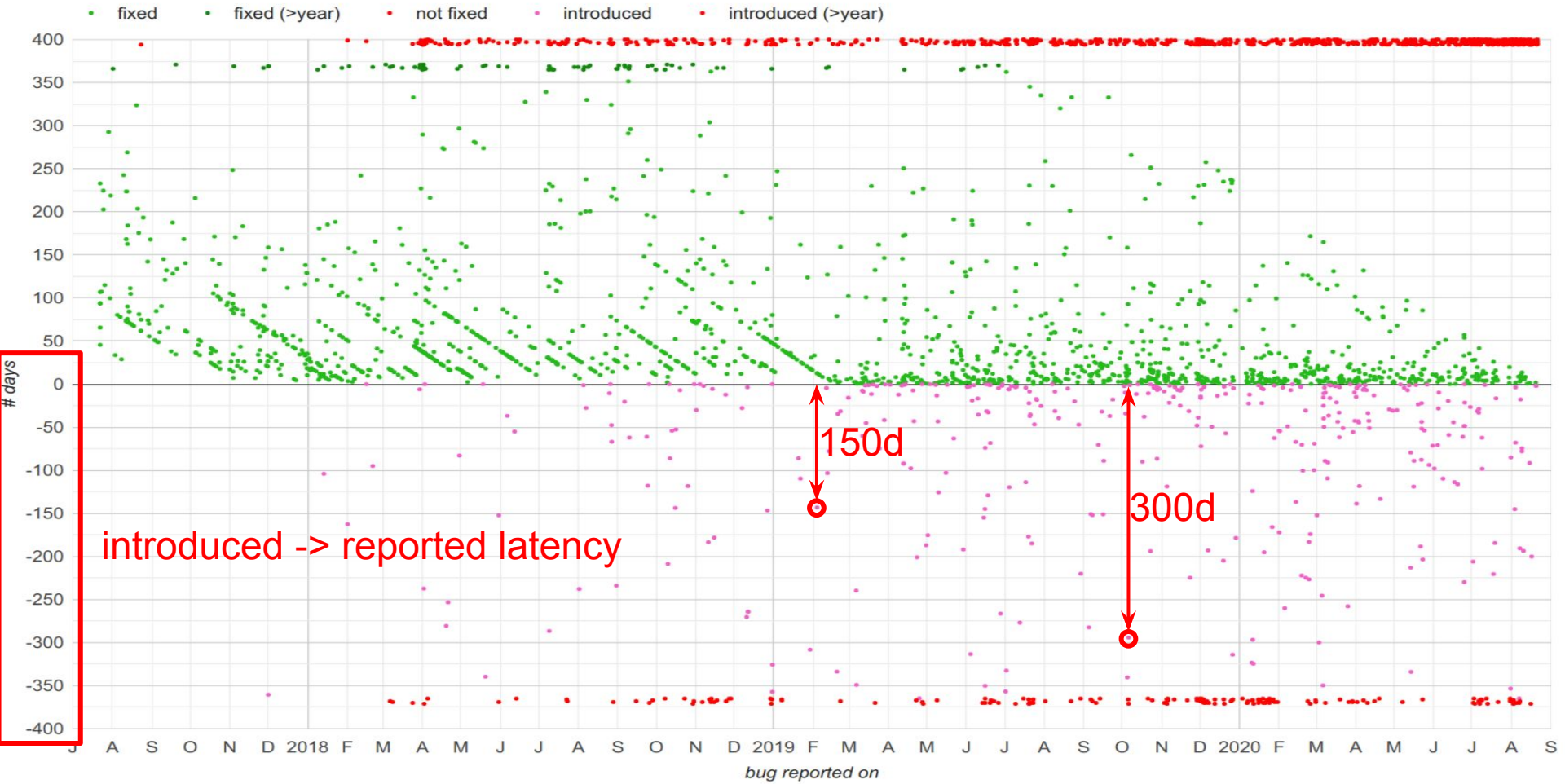
Bug Lifetimes



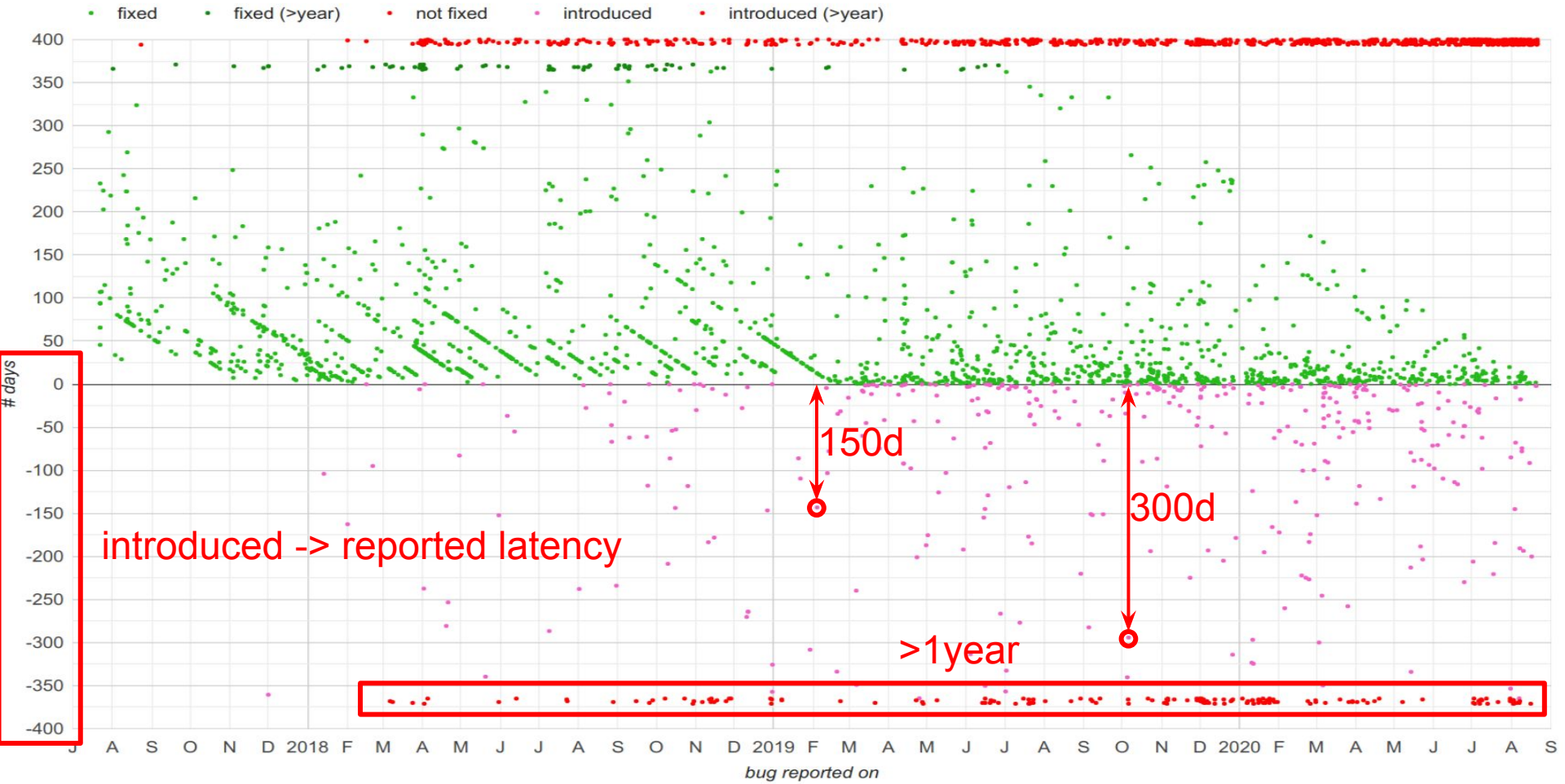
Bug Lifetimes



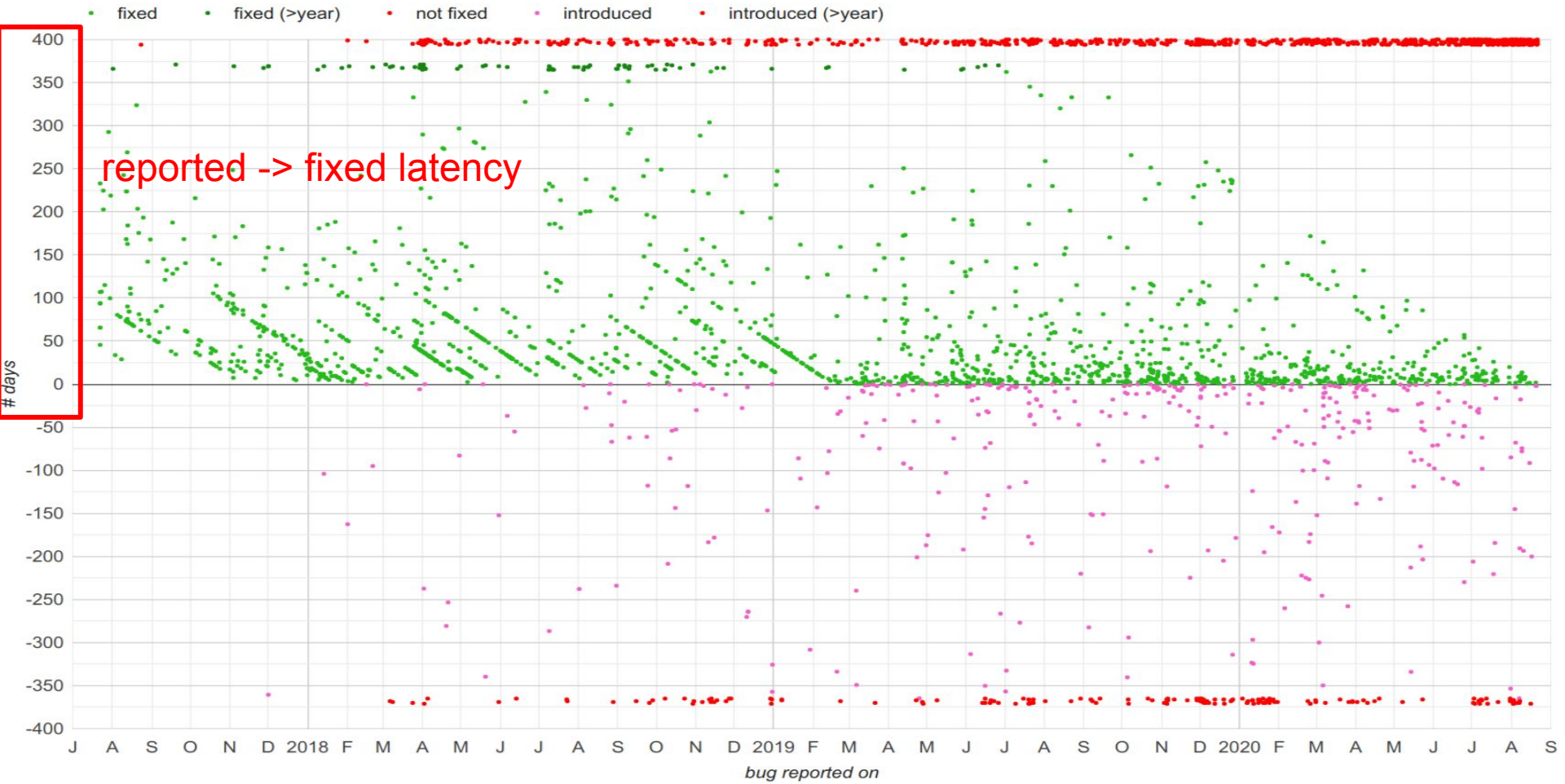
Bug Lifetimes



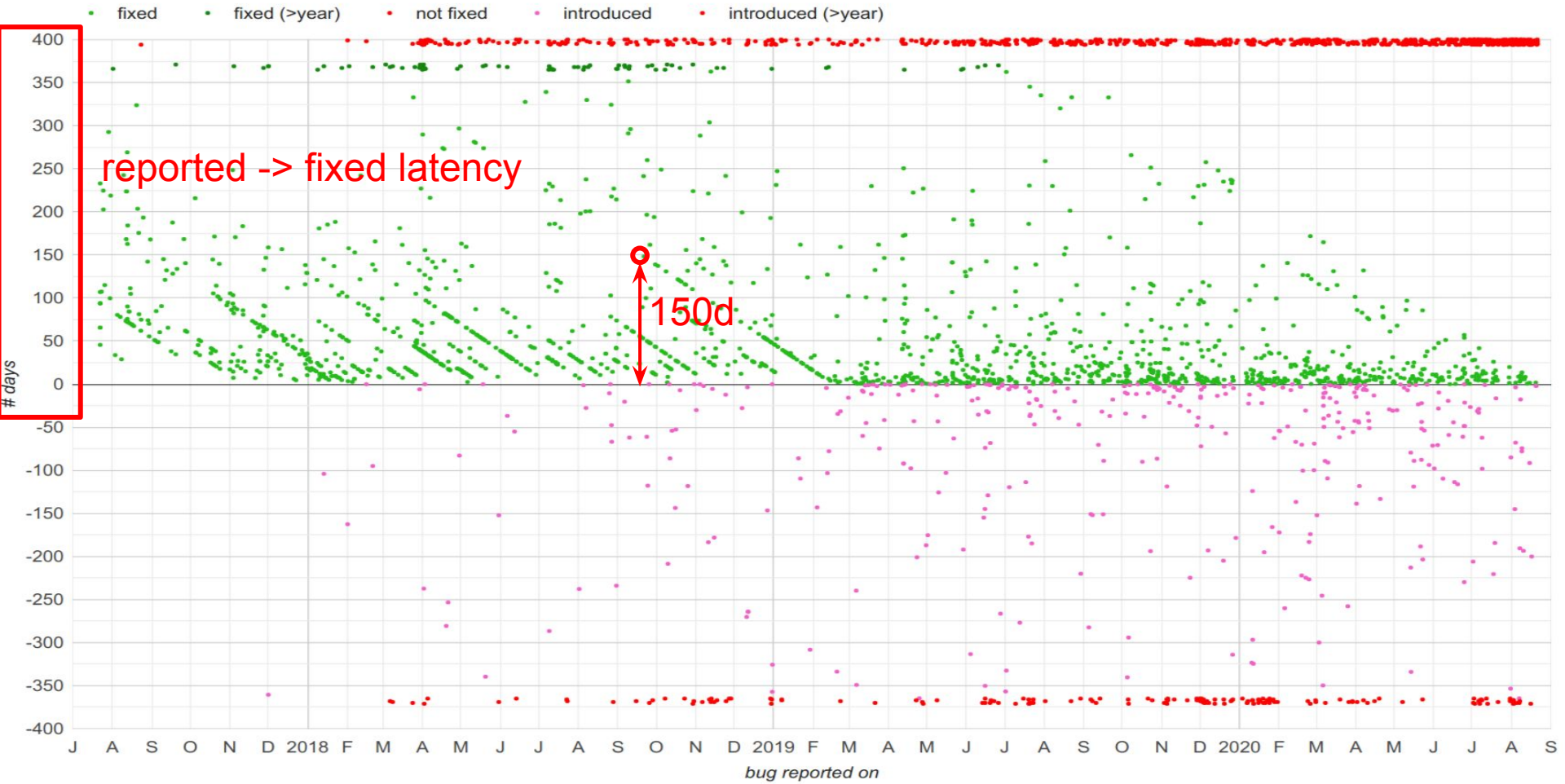
Bug Lifetimes



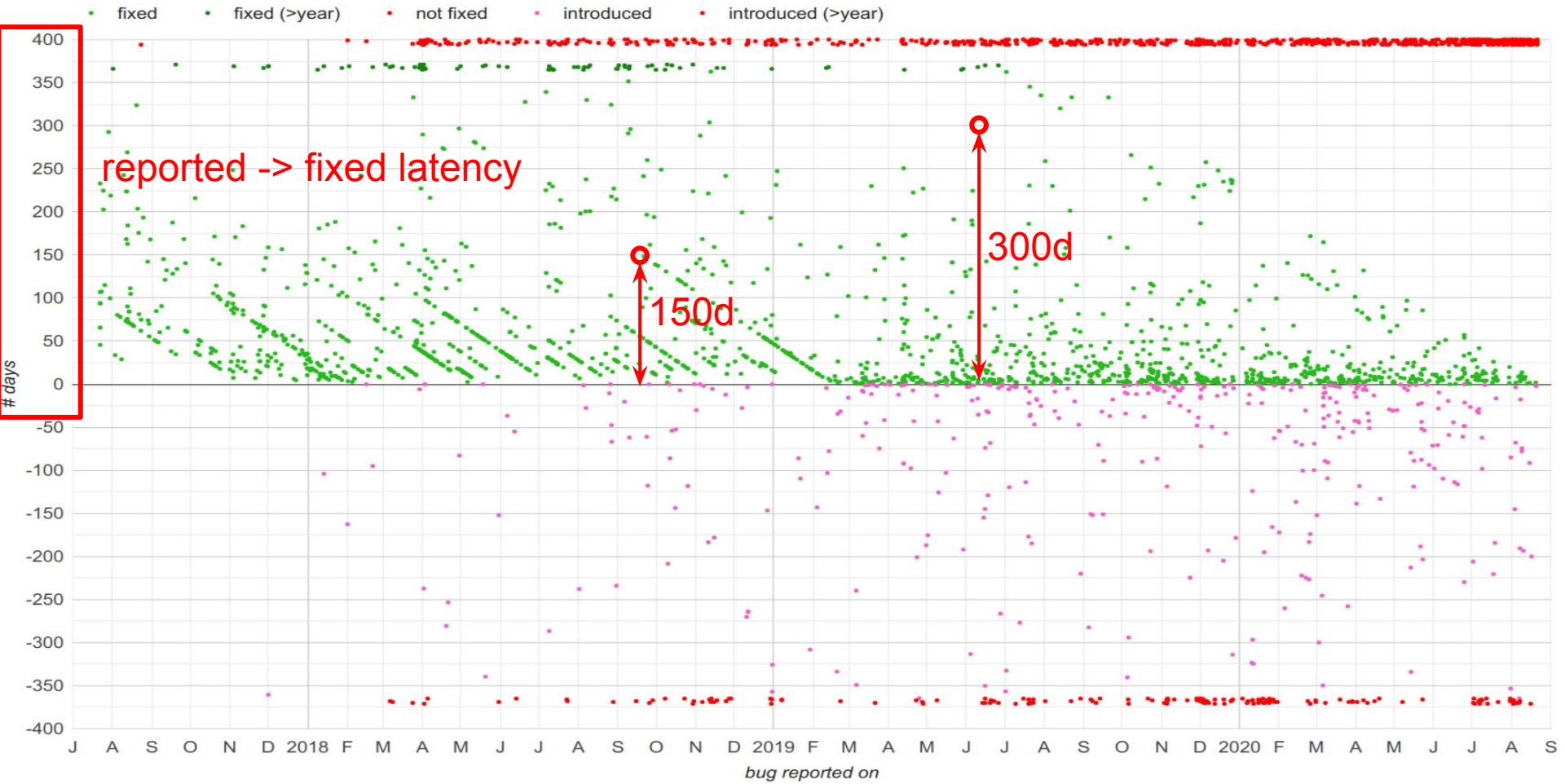
Bug Lifetimes



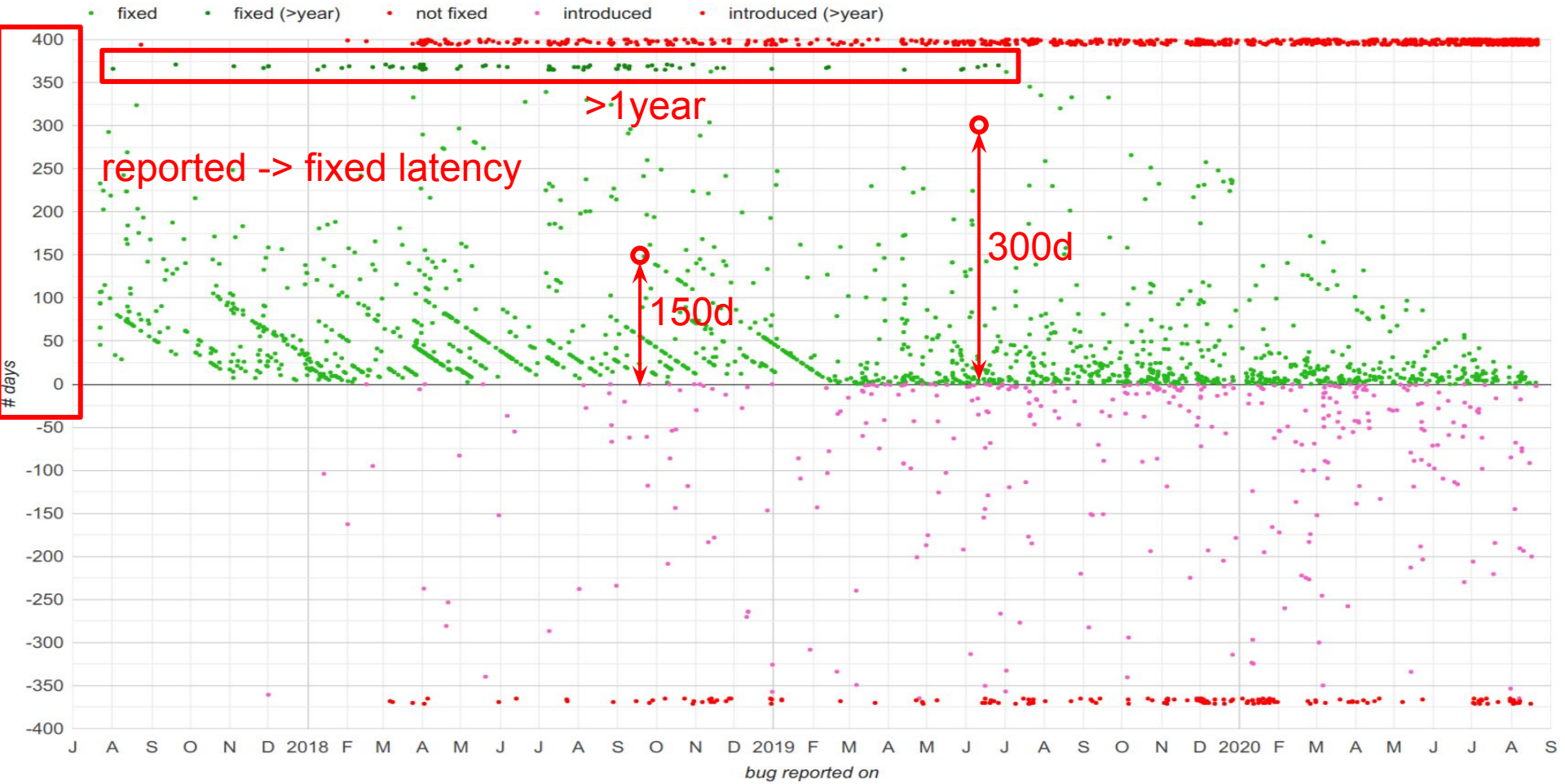
Bug Lifetimes



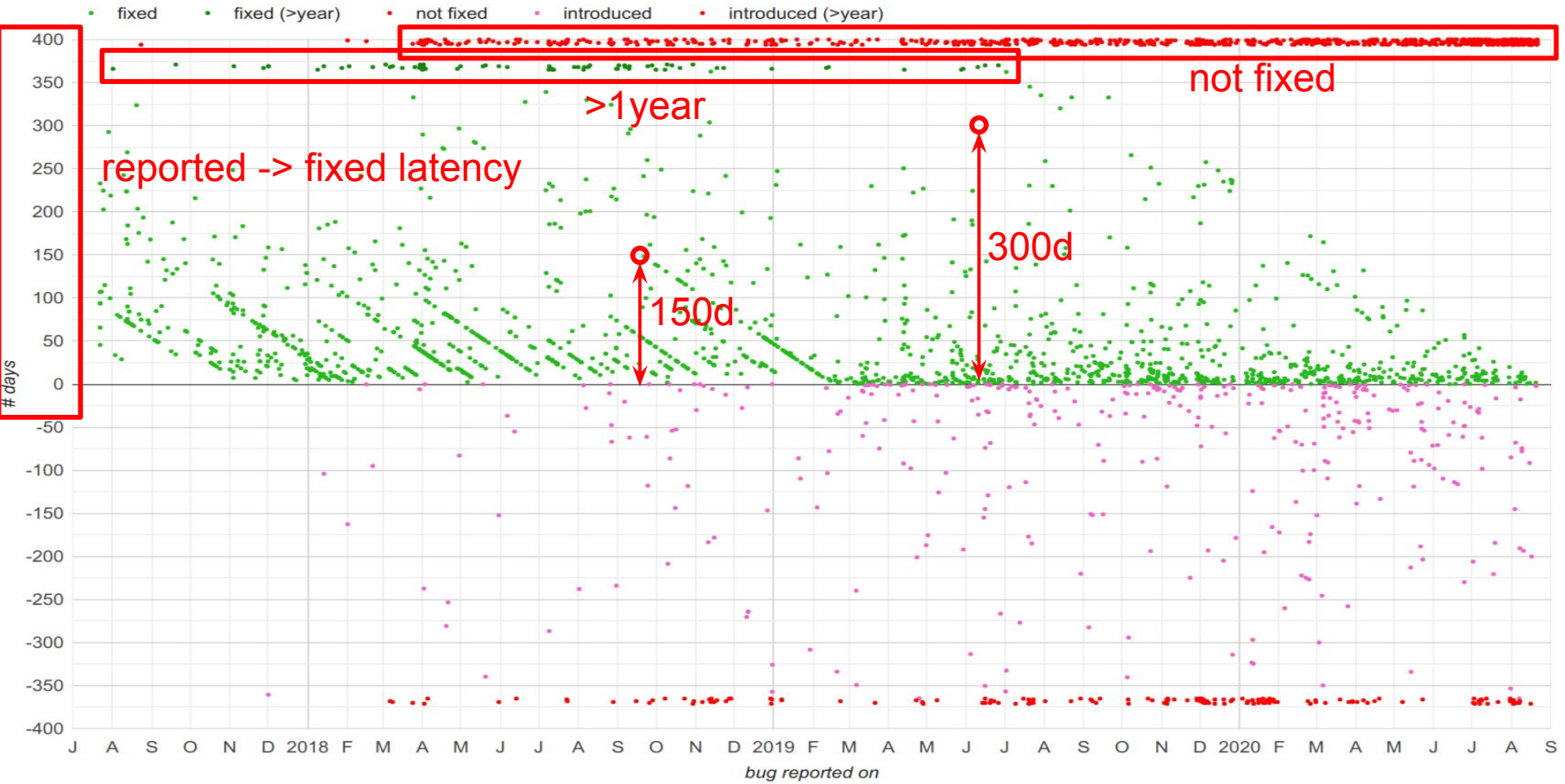
Bug Lifetimes



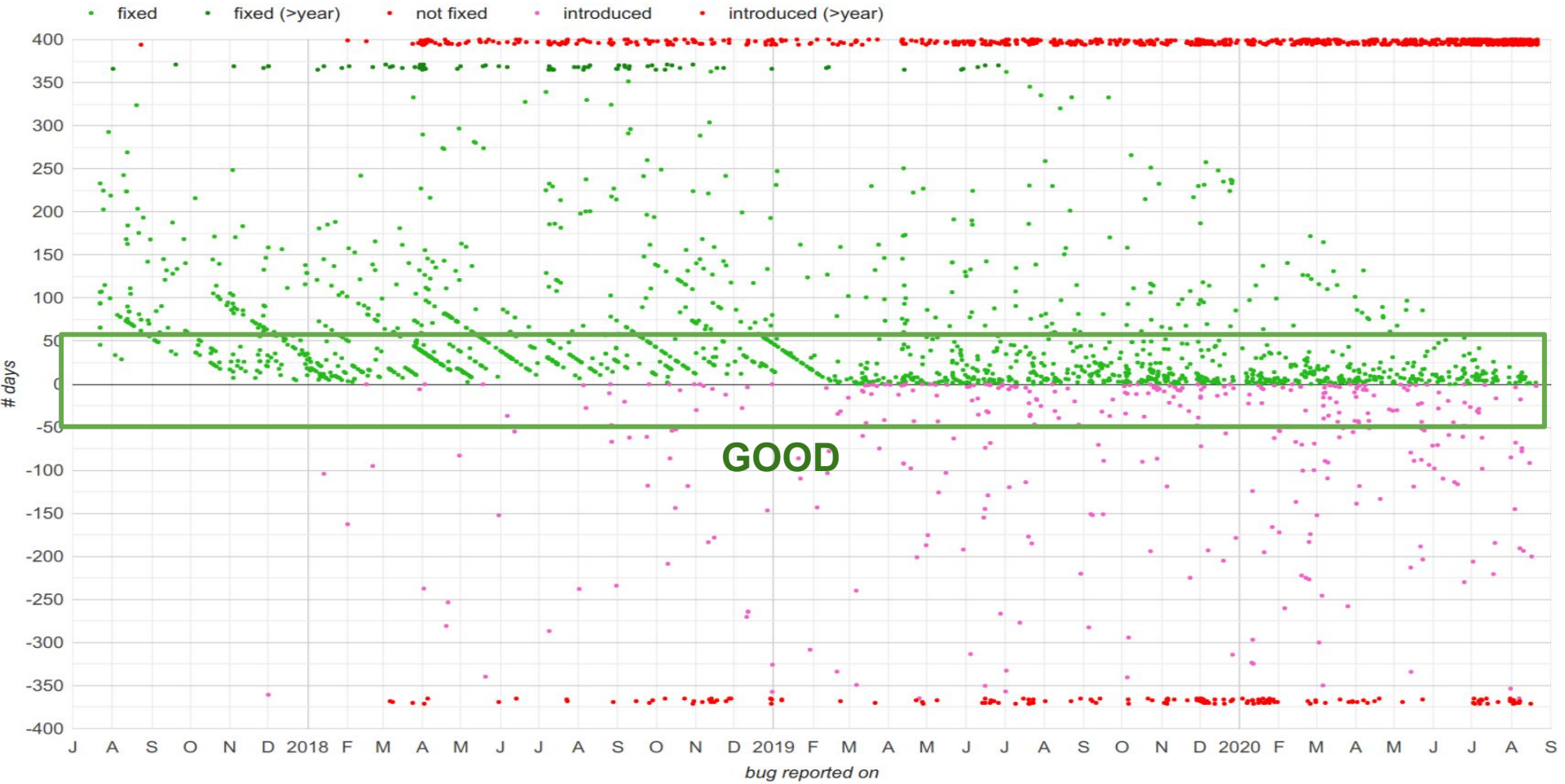
Bug Lifetimes



Bug Lifetimes



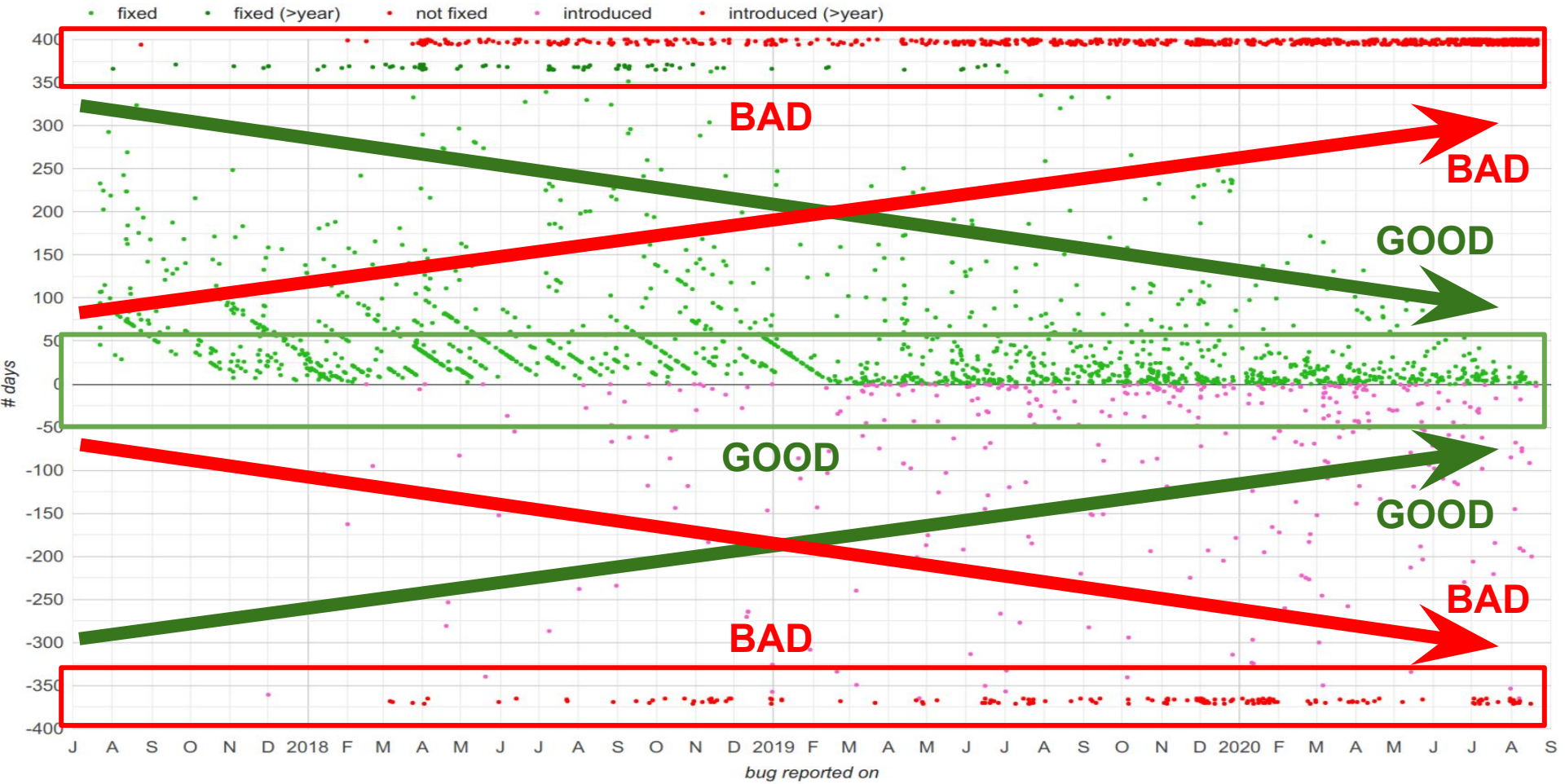
Bug Lifetimes



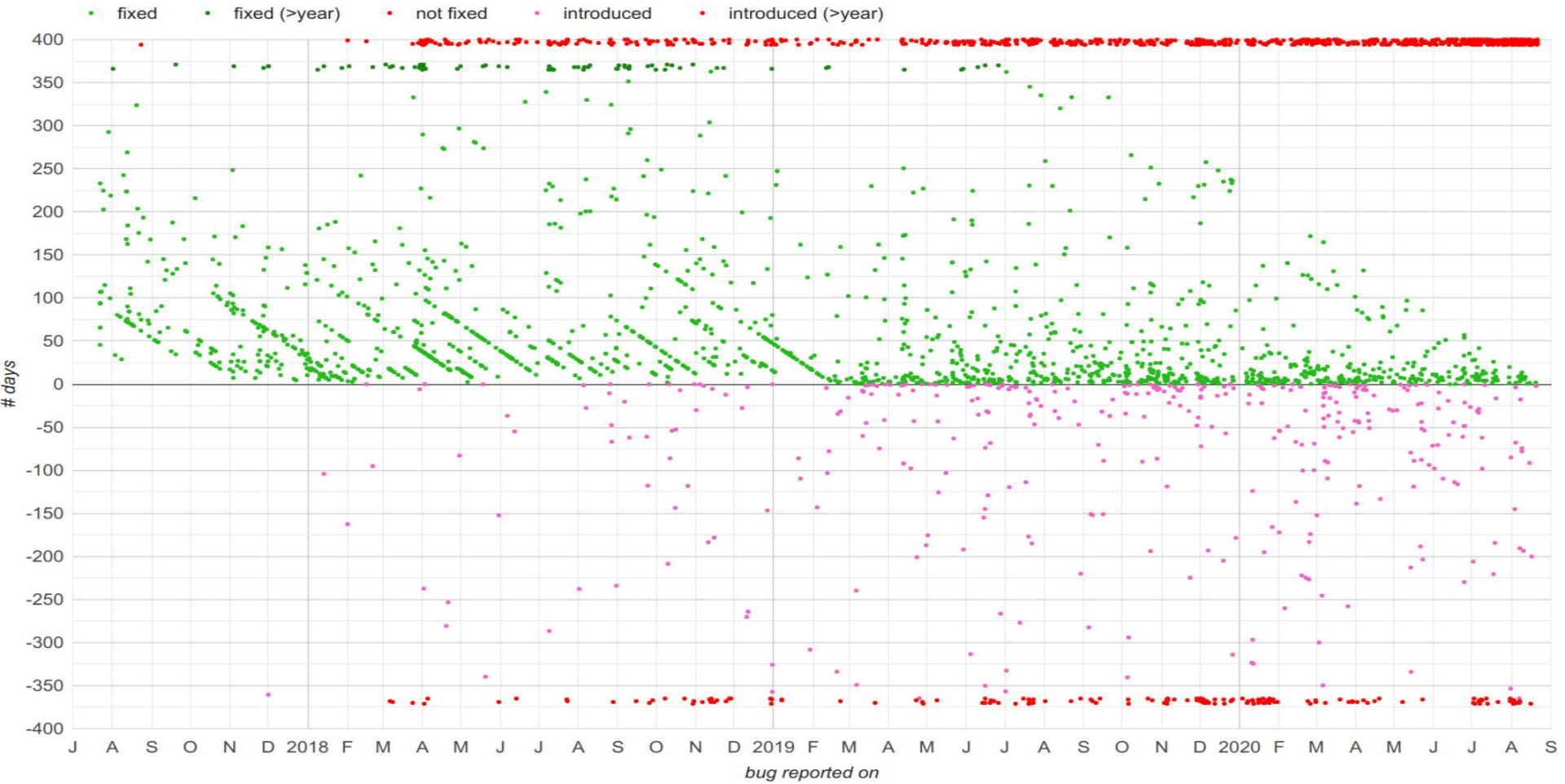
Bug Lifetimes

Bug Lifetimes

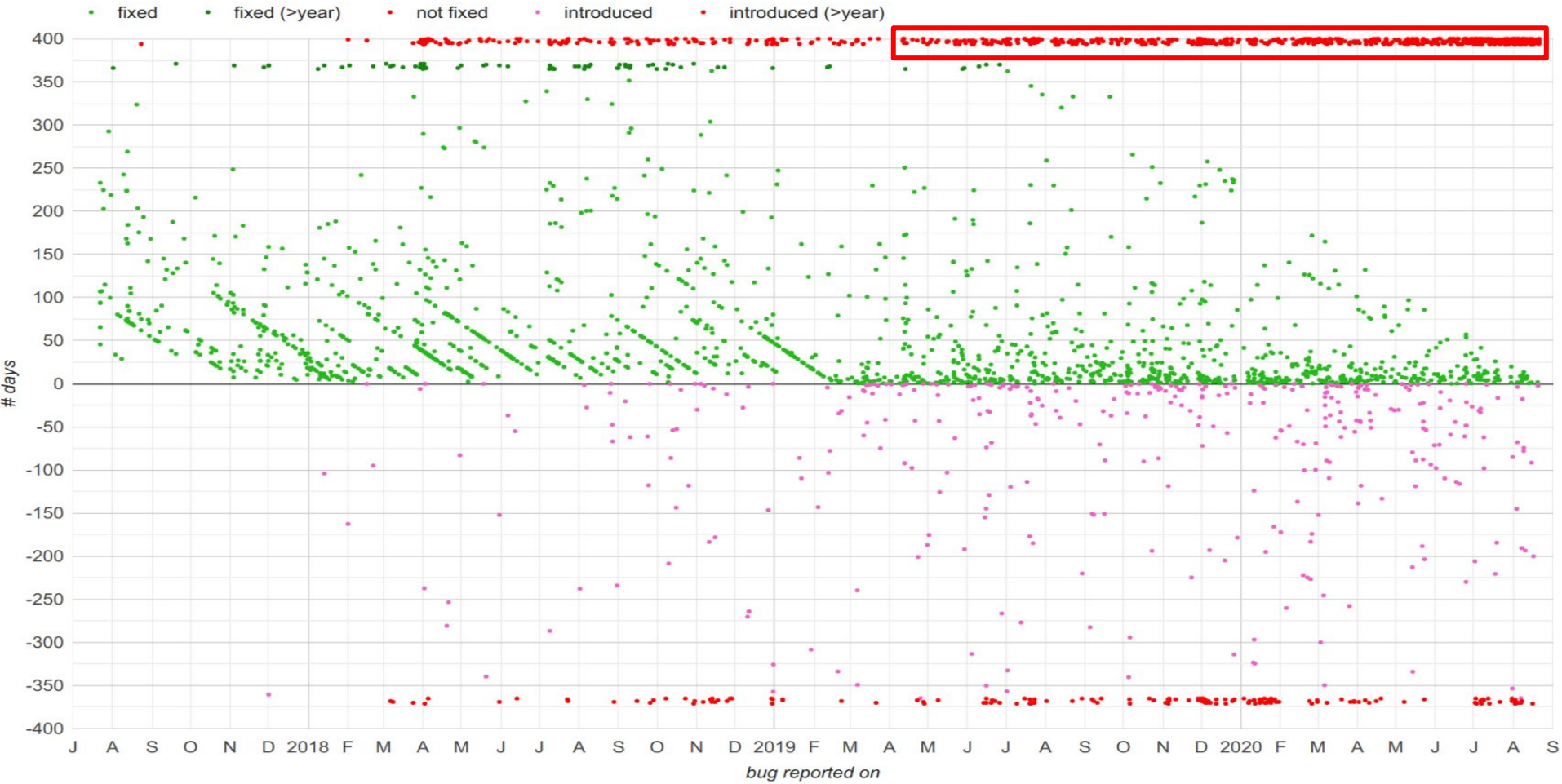
Bug Lifetimes



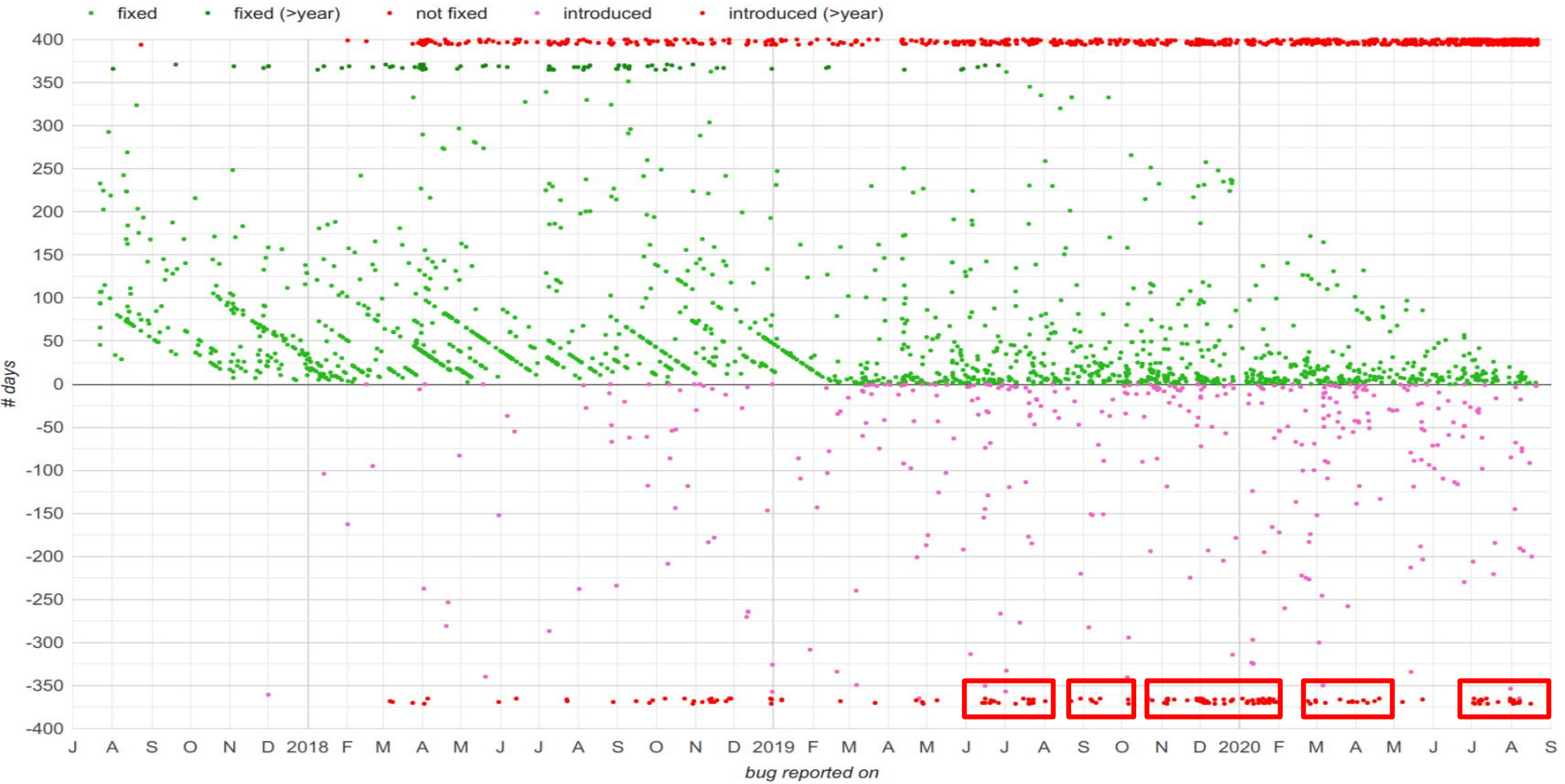
Bug Lifetimes



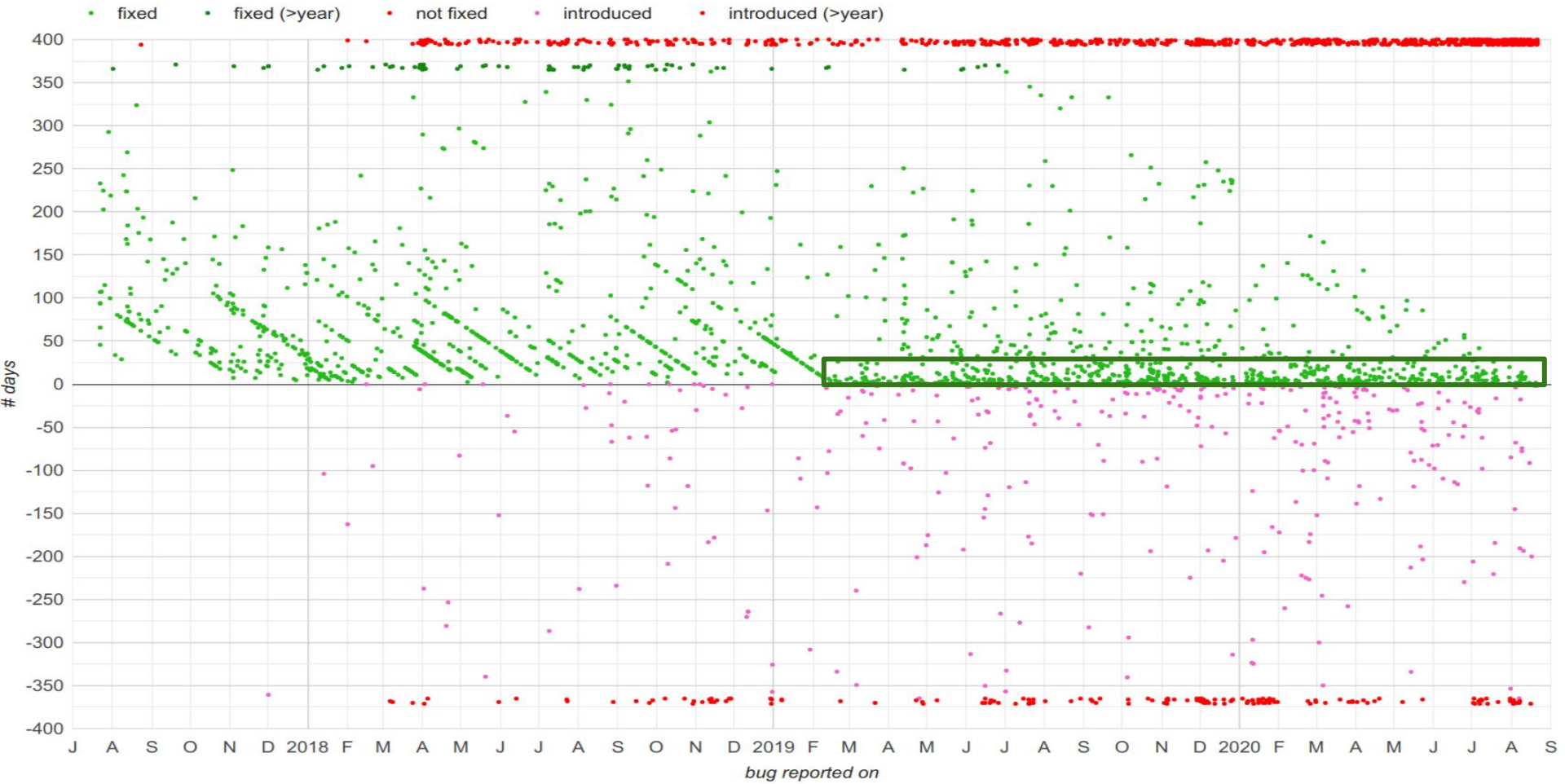
Bug Lifetimes



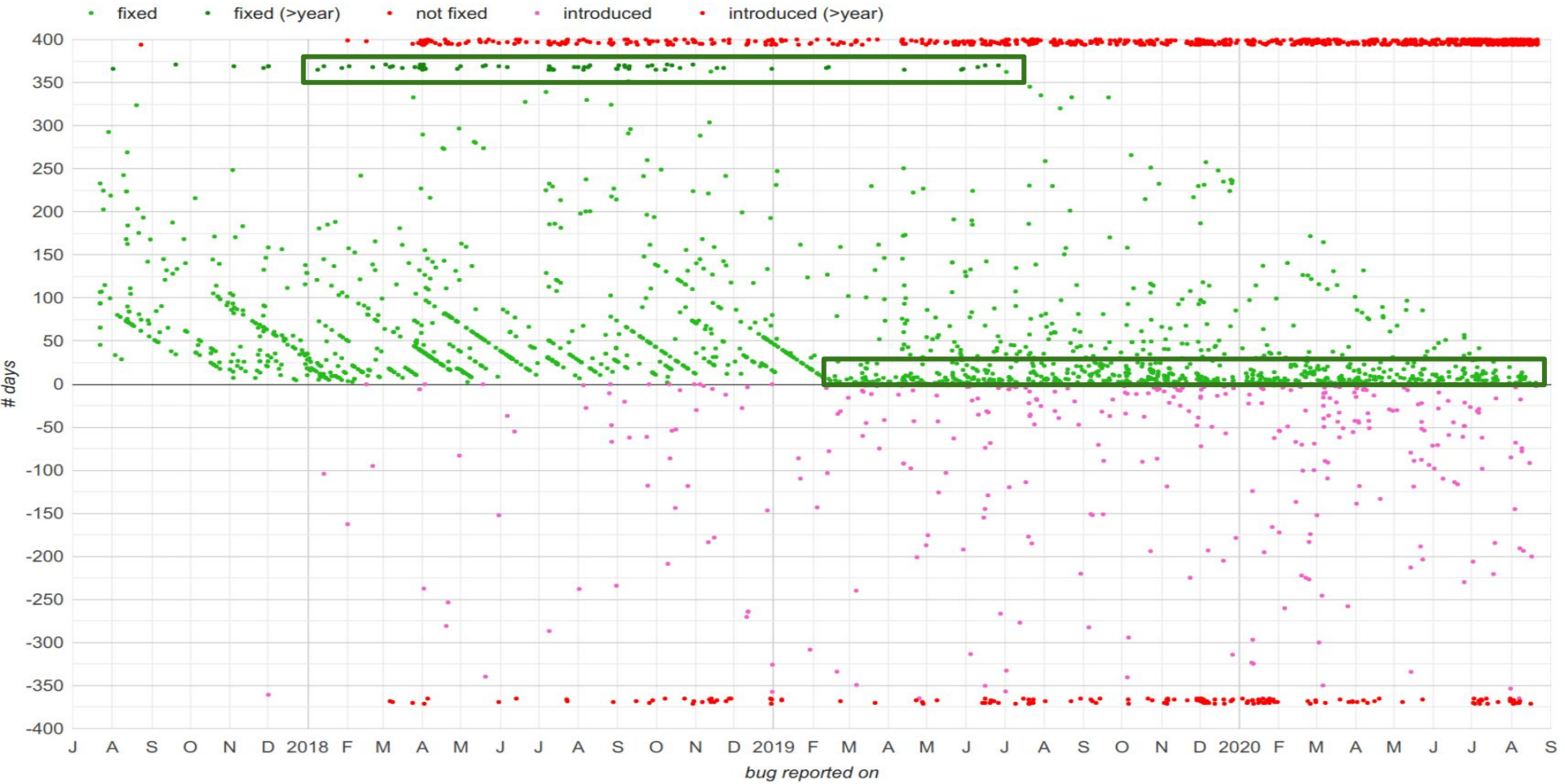
Bug Lifetimes



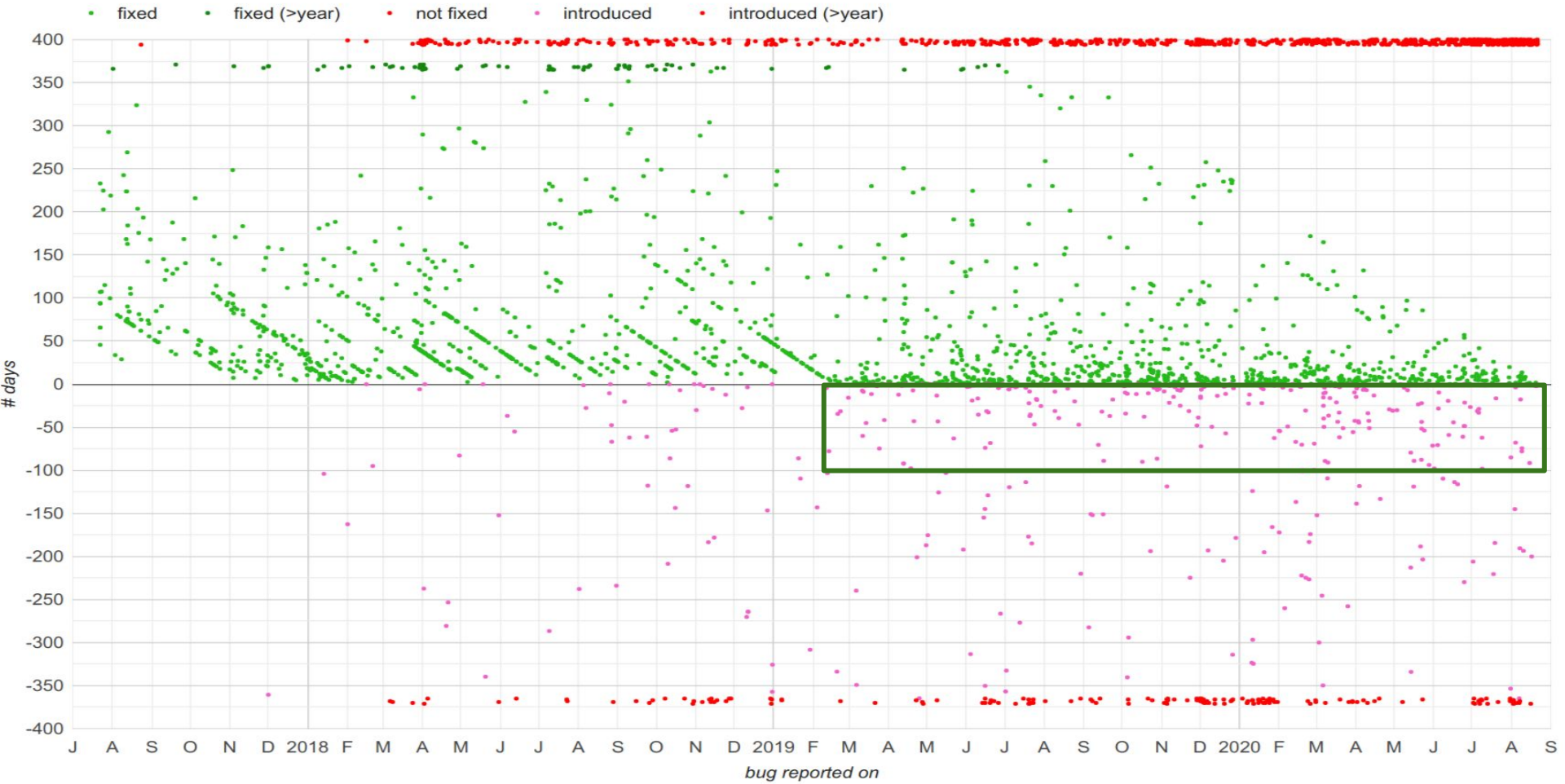
Bug Lifetimes



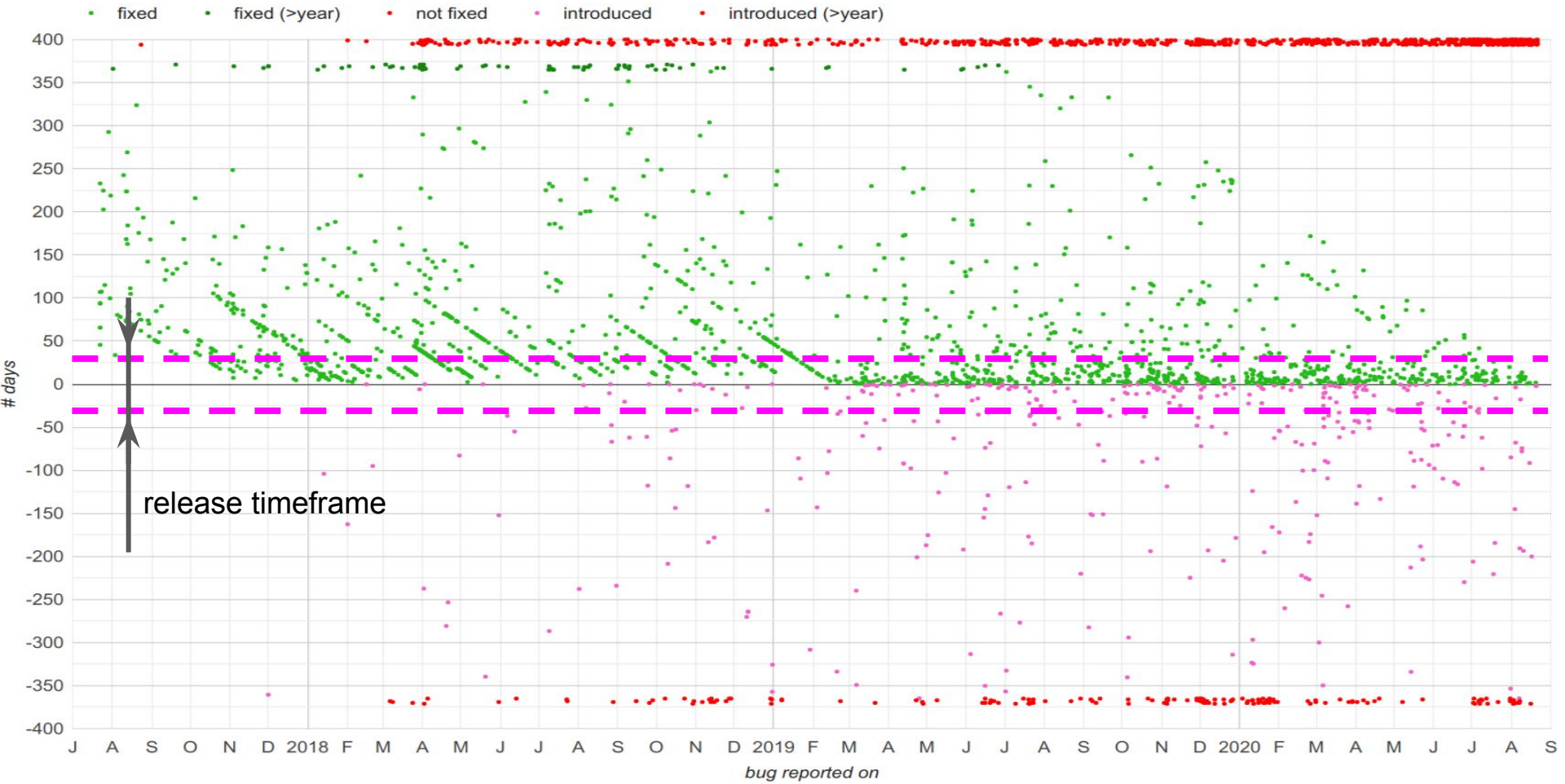
Bug Lifetimes



Bug Lifetimes



Bug Lifetimes



Thank you!

Q&A

syzkaller@googlegroups.com
kasan-dev@googlegroups.com

Dmitry Vyukov (dvyukov@)

LTS

	Open	"Fixed"	"Obsolete"
4.19	368	140	189
4.14	506	100	22
4.9	WARNING in cpumask_check		
4.4	WARNING in batadv_tvlv_container_remove		