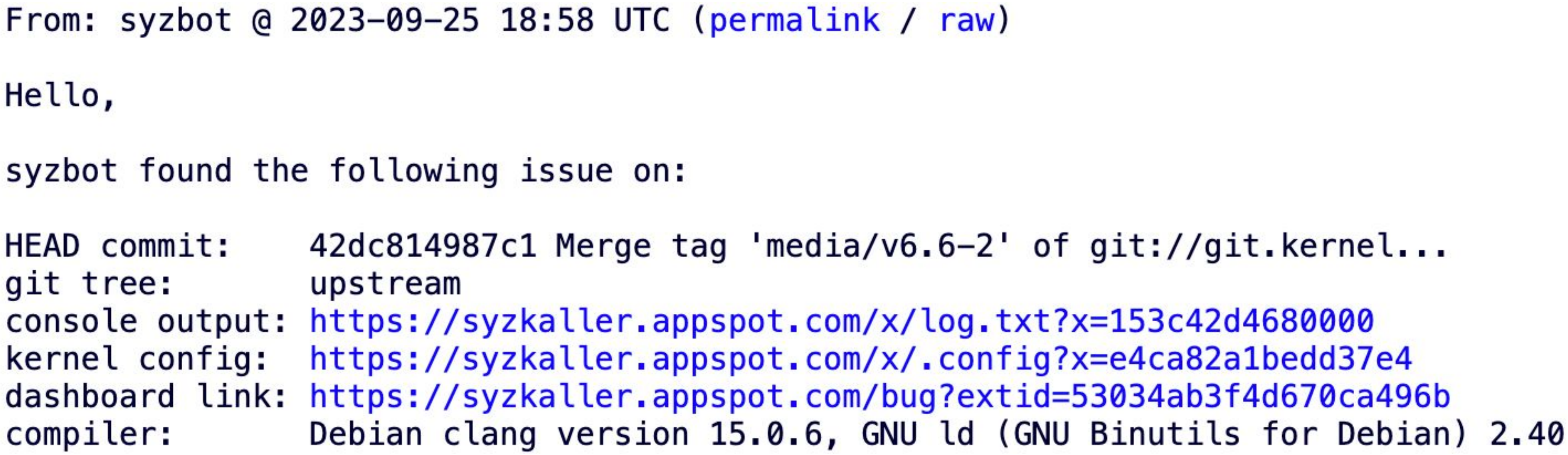
Syzbot BoF (LPC'24)

Hosted by: Aleksandr Nogikh (Google),
Alexander Potapenko (Google), Dmitry Vyukov (Google),
Taras Madan (Google)

Syzkaller and Syzbot

- **syzkaller** (coverage-guided kernel fuzzer) appeared in **2015.**
  - Syzkaller is a standalone application.
- **syzbot** has begun to report kernel findings to LKML in **2017.**
  - Syzbot is a continuous kernel build / fuzz / report aggregation system.
  - Syzbot uses **syzkaller** for the actual fuzzing.
- **~12.3k** findings have been uncovered over the years.
- **~4.8k** Linux kernel commits directly mention syzbot or syzkaller.

E-Mail Reports

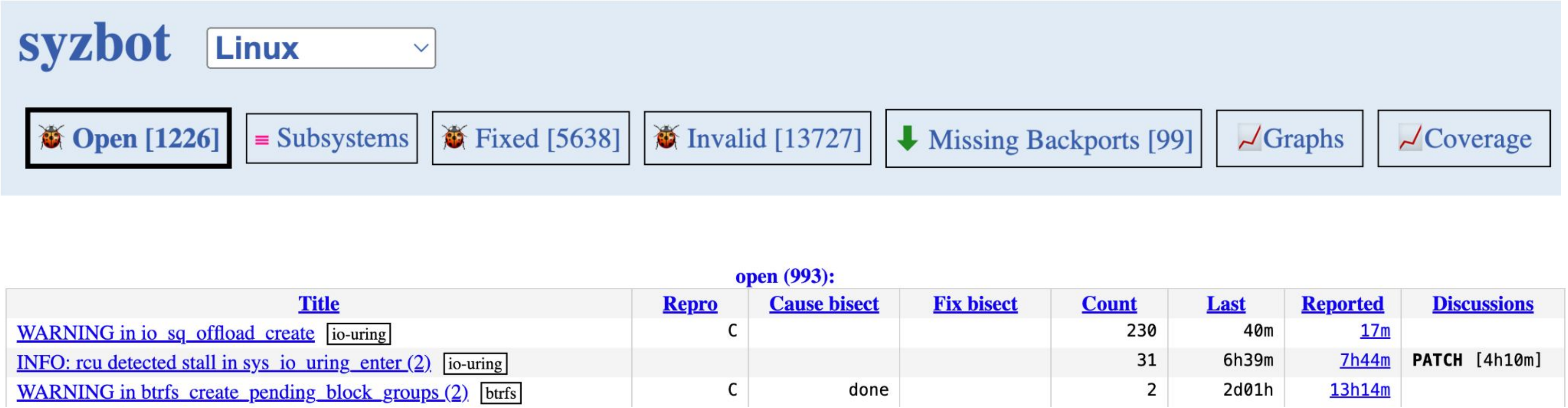


```
From: syzbot @ 2023-09-25 18:58 UTC (permalink / raw)

Hello,

syzbot found the following issue on:

HEAD commit:    42dc814987c1 Merge tag 'media/v6.6-2' of git://git.kernel...
git tree:       upstream
console output: https://syzkaller.appspot.com/x/log.txt?x=153c42d4680000
kernel config:  https://syzkaller.appspot.com/x/.config?x=e4ca82a1bedd37e4
dashboard link: https://syzkaller.appspot.com/bug?extid=53034ab3f4d670ca496b
compiler:       Debian clang version 15.0.6, GNU ld (GNU Binutils for Debian) 2.40
```

< ... >

Web Dashboard

https://syzkaller.appspot.com/upstream



open (993):

Title	Repro	Cause bisect	Fix bisect	Count	Last	Reported	Discussions
WARNING in io_sq_offload_create io-uring	C			230	40m	17m	
INFO: rcu detected stall in sys_io_uring_enter (2) io-uring				31	6h39m	7h44m	PATCH [4h10m]
WARNING in btrfs_create_pending_block_groups (2) btrfs	C	done		2	2d01h	13h14m	

Syzbot in 2024 (Jan-Aug)

1479 reported bugs

(**944** during Jan-Aug 2023)

1127 tested fix candidates

(**750** during Jan-Aug 2023)

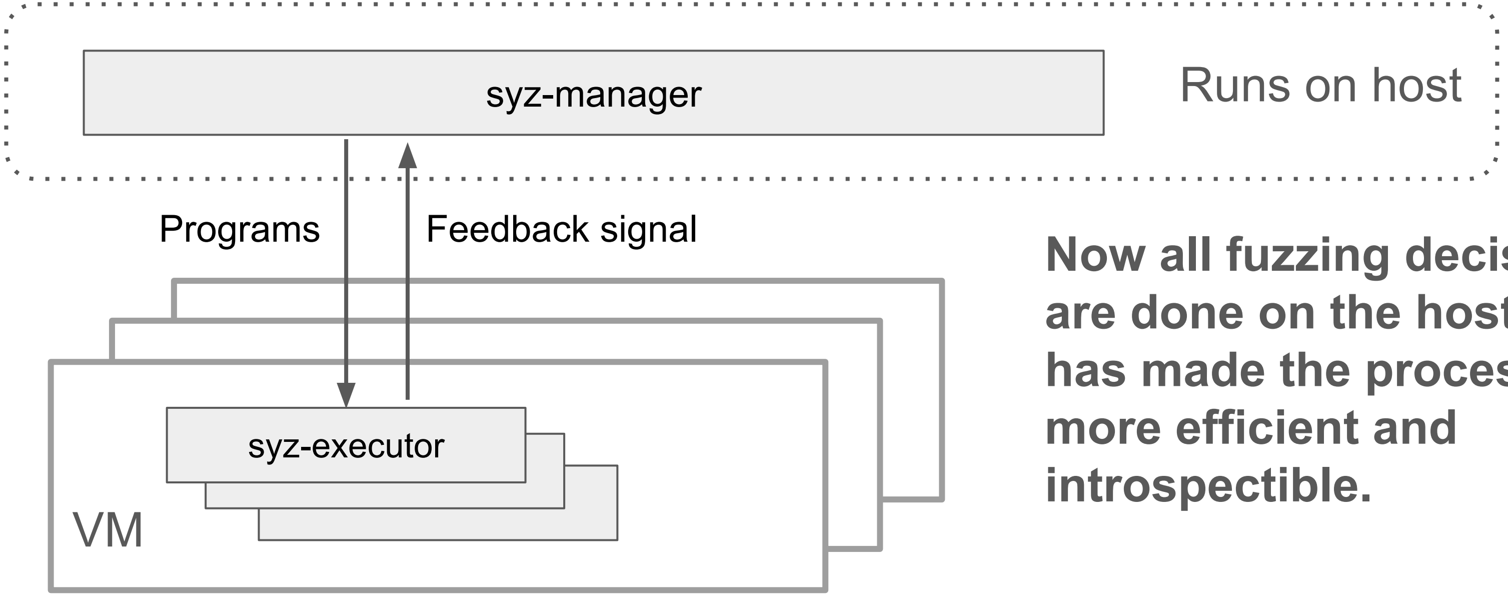
530 fixes for reported bugs

(**479** in Jan-Aug 2023)

...but still **1226** open bugs :(

https://syzkaller.appspot.com/upstream

Fuzzing Engine Refactoring(s)



Now all fuzzing decisions are done on the host, which has made the process much more efficient and introspectible.

Fuzzing decisions used to be done independently inside each VM

Snapshot-based Fuzzing

The implementation is based on QEMU's loadvm/savevm.

Boot a VM -> **take a snapshot** (1) -> **execute a program** -> **rollback to** (1)

The objective was to make kernel fuzzing as side-effect free as possible:

- (Hopefully) Achieve a more stable coverage of the kernel.
- (Hopefully) Improve bug reproducibility.

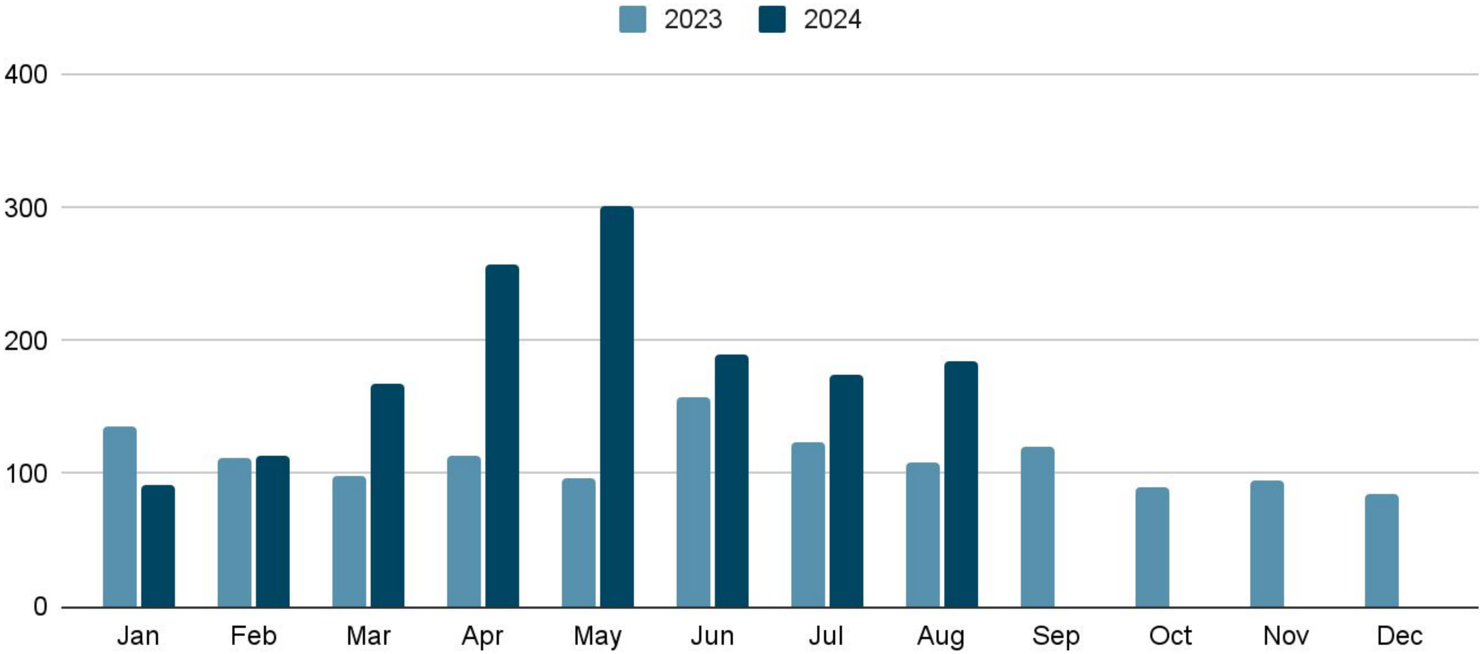
Snapshot-based Fuzzing: 1 Month Results

- The coverage is **3.6%** higher than on other clang-based instances.
- **60+** bugs that were detected only on the snapshot-based instance.
- **~75%** bugs have a reproducer compared to **~40%** of bugs from other instances.

Kernel Code Coverage [2024]

- Newly fuzzed subsystems:
  - bcachefs (144 findings!)
- Improvements:
  - BPF descriptions [see the LPC talk by Paul Chaignon]
  - KVM descriptions
- Ongoing effort:
  - Automated description generation based on static analysis.
    - But it's unlikely to work well for nontrivial kernel interfaces.

2023 vs 2024



^ the start of fuzzing engine improvements

Missing Backports

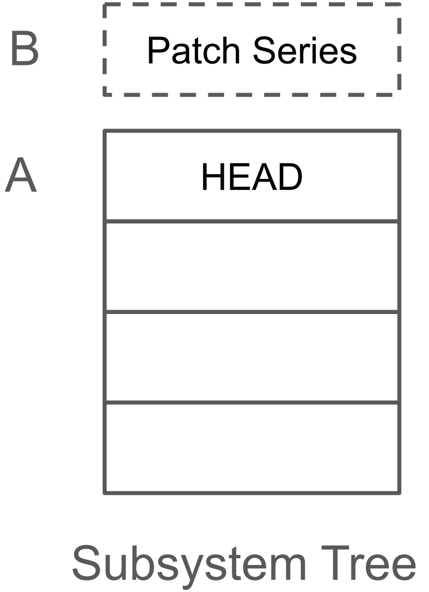
https://syzkaller.appspot.com/upstream/backports

Syzbot has already detected more than **100** commits that *very likely* address Linux LTS kernels bugs.

Many of those bugs (**70+%**) do no apply cleanly, but the conflicts are minimal.

We're currently preparing an experimental batch of backports to figure out the right filtering/preprocessing approaches.

Next: Patch Fuzzing (currently WIP)



B

Patch Series

A

HEAD

Subsystem Tree

Focused Fuzzing of B

Find the reproducers

Run the repro on A

A not crashed => The patch has a bug

Relative to the bugs later reported by syzbot, **precision** (>95%) and **recall** (up to 60-70%) figures look promising.

We need your feedback and cooperation

- To find more bugs, syzbot needs some human aid.
    - You may contribute to the syzlang descriptions of your kernel subsystem's interface.
    - Adding more assertions and self-checking functionality helps detect more bugs.
    - Fixing the currently open bugs helps the fuzzer uncover deeper problems in the code.
- You can influence what syzbot reports and what it does not.
    - Do you see any repetitive cases of false positive/irrelevant reports?
    - What extra information could help you triage and debug the reports faster?
- **Don't hesitate to reach out to** syzkaller@googlegroups.com