# Syzbot BoF (2025)

**Google:**
Aleksandr Nogikh
Alexander Potapenko
Dmitry Vyukov
Marco Elver
Taras Madan

# Agenda

- Statistics (2025)
- Updates (2025)
  - Domain
  - Patch Fuzzing
  - Coverage Aggregation
  - Rust Code Fuzzing
  - KVM
  - KFuzzTest
- 2026
  - Revisit Email Reporting
  - Patch Fuzzing (cont)
  - AI + Patch Generation & Code Reviews

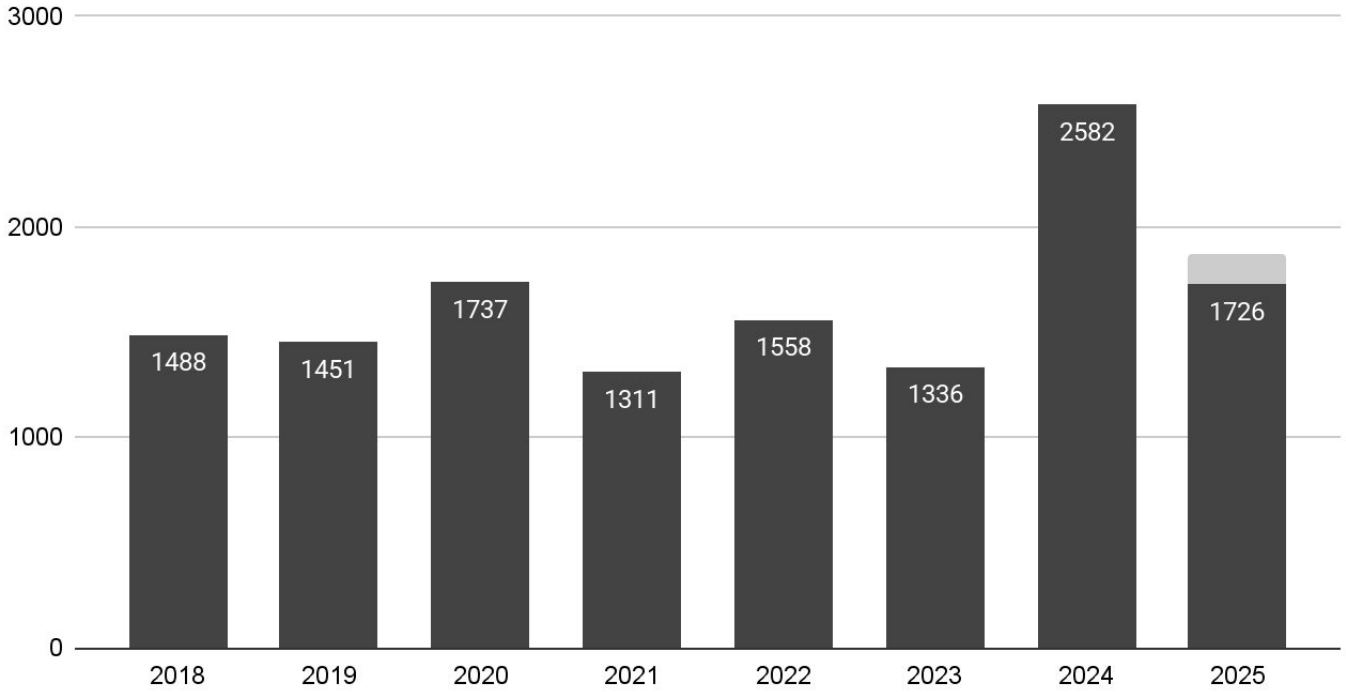
# Context

**syzkaller** is a coverage-guided kernel fuzzer.

**syzbot** is a continuous kernel build / fuzz / report aggregation system.
**syzbot** uses **syzkaller** for the actual fuzzing.

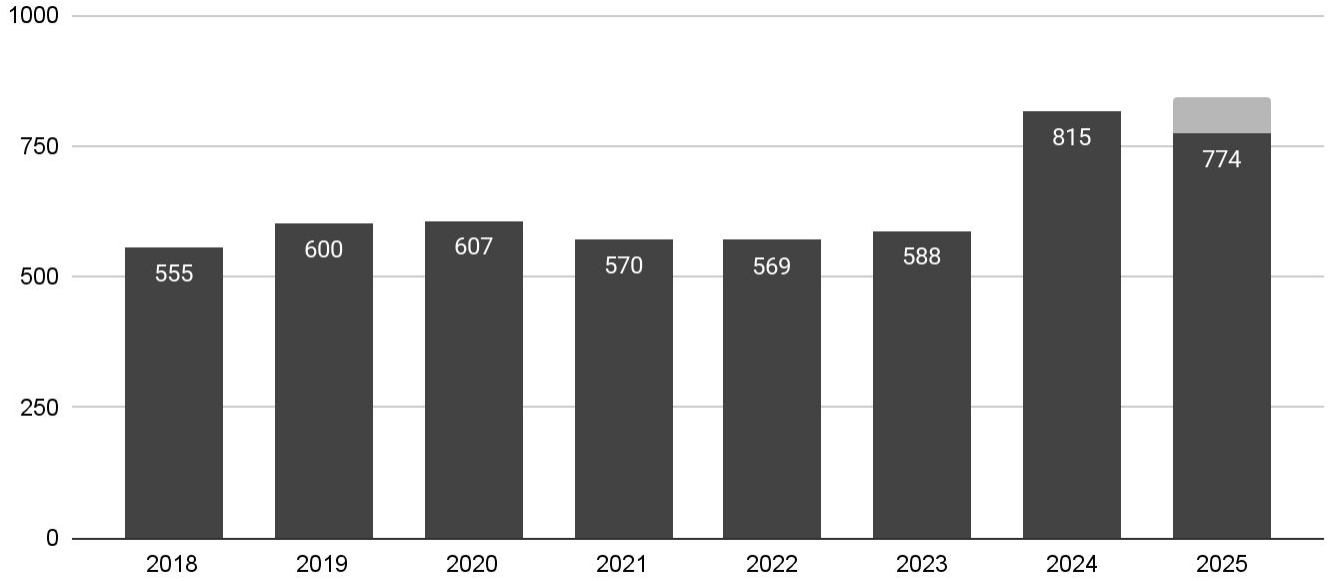
Since 2017, syzbot has reported more than **13,500** bugs to the mailing lists.

# Reported Findings (Yearly)

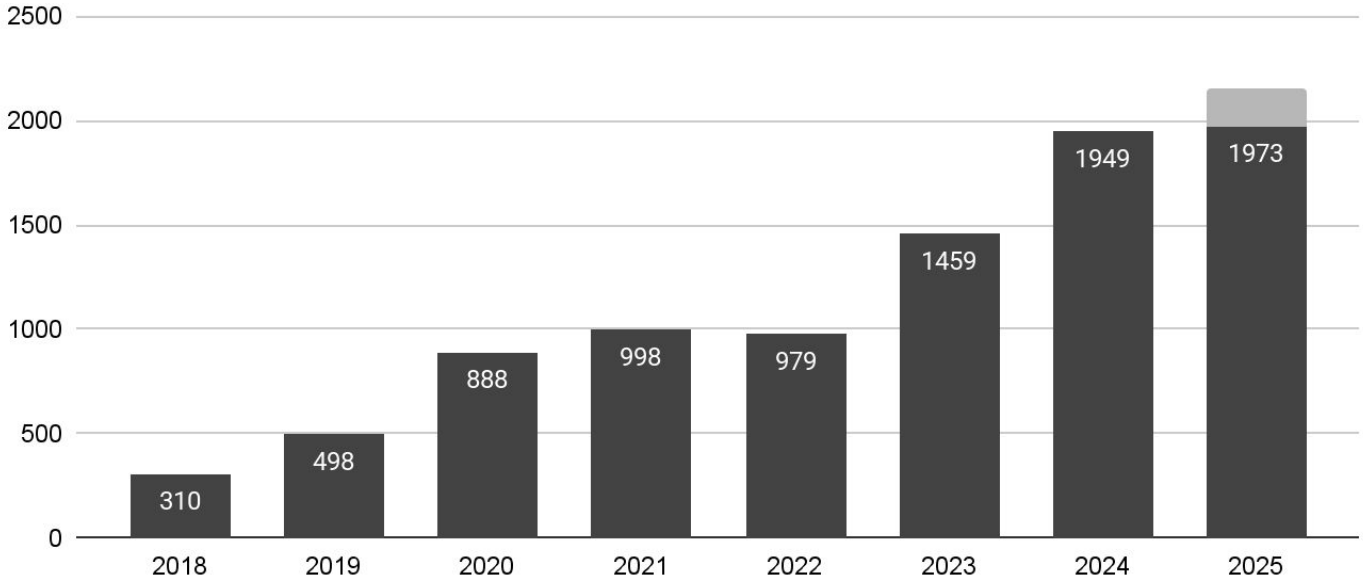


# Bug Fixes (Yearly)

Yearly kernel commits that mention syzbot or syzkaller.



# Patch Testing Requests (Yearly)



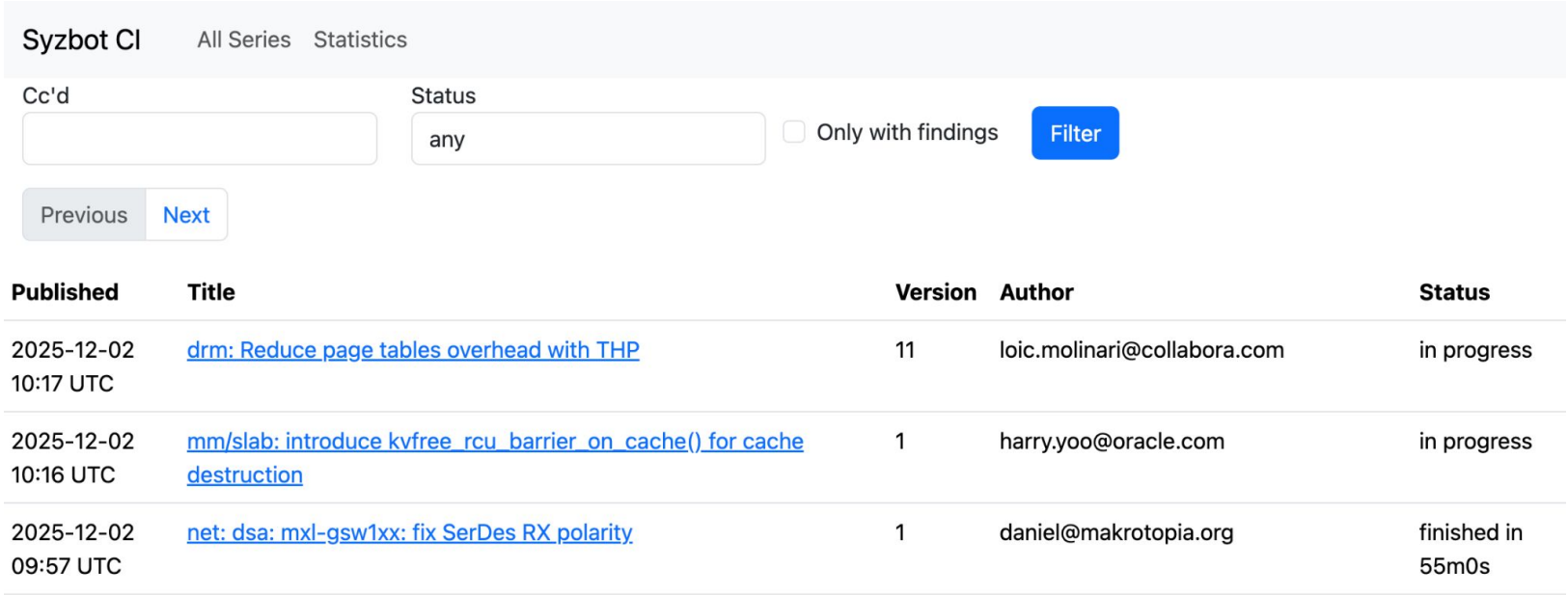
# Updates (2025)

# New Domain!

We're now also available at:
https://syzbot.org/

# Patch Fuzzing ("syzbot CI")

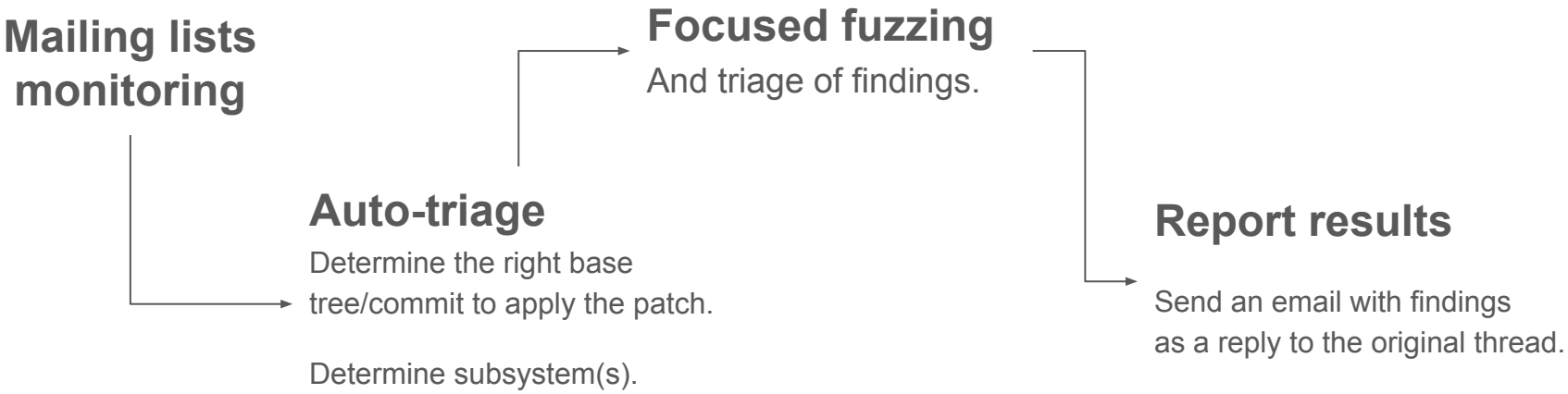
https://ci.syzbot.org




| Published | Title | Version | Author | Status |
|---|---|---|---|---|
| 2025-12-02 10:17 UTC | drm: Reduce page tables overhead with THP | 11 | loic.molinari@collabora.com | in progress |
| 2025-12-02 10:16 UTC | mm/slab: introduce kvfree_rcu_barrier_on_cache() for cache destruction | 1 | harry.yoo@oracle.com | in progress |
| 2025-12-02 09:57 UTC | net: dsa: mxl-gsw1xx: fix SerDes RX polarity | 1 | daniel@makrotopia.org | finished in 55m0s |

There's also a separate syzbot CI talk at the Kernel Testing & Dependability MC today.

# Patch Fuzzing: High-Level Approach



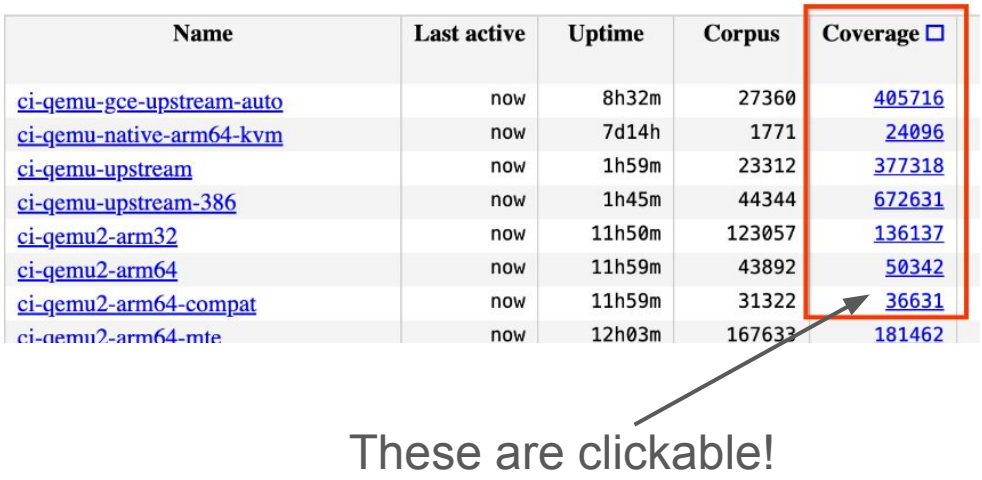

**Objective:** share results before series are applied to maintainer trees.

# Patch Fuzzing: Questions / Discussion

- Have you already encountered "Syzbot CI" reports?
  What is missing / what can be improved?
- It's possible to add more steps to the processing pipeline (e.g. run some test utilities).
- **To maintainers:** how would such a pre-merge fuzzing best fit into your workflow?

# Coverage Reports

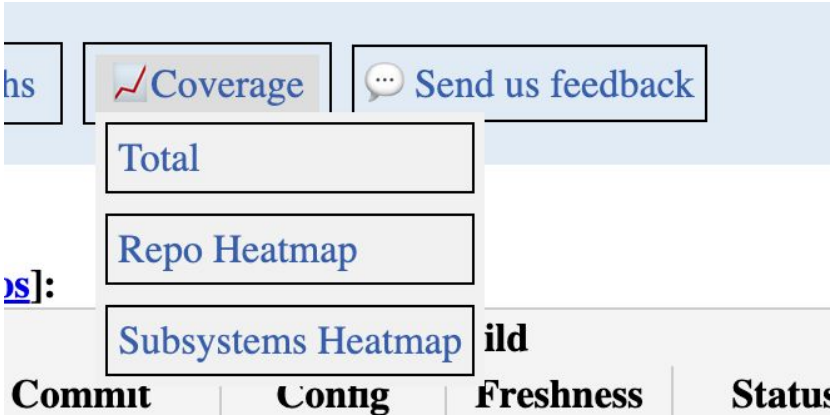
Heavy programs are now excluded from the coverage reports, so these reports have reasonable size again.



| Name | Last active | Uptime | Corpus | Coverage |
|---|---|---|---|---|
| ci-qemu-gce-upstream-auto | now | 8h32m | 27360 | 405716 |
| ci-qemu-native-arm64-kvm | now | 7d14h | 1771 | 24096 |
| ci-qemu-upstream | now | 1h59m | 23312 | 377318 |
| ci-qemu-upstream-386 | now | 1h45m | 44344 | 672631 |
| ci-qemu2-arm32 | now | 11h50m | 123057 | 136137 |
| ci-qemu2-arm64 | now | 11h59m | 43892 | 50342 |
| ci-qemu2-arm64-compat | now | 11h59m | 31322 | 36631 |
| ci-qemu2-arm64-mte | now | 12h03m | 167633 | 181462 |

# Aggregated Coverage Reports

The web dashboard now displays the coverage report that is aggregated across all syzkaller instances fuzzing a particular kernel.




It can be used to identify coverage gaps for particular kernel subsystems.

# Rust Code Fuzzing

**New:** support for Rust code fuzzing.

- CONFIG_KCOV works for the Rust code (kudos to Alice Ryhl!)
- Syzkaller can generate coverage reports for the Rust code.
- Syzbot can now compile CONFIG_RUST=y kernels.

**First results:**
8 findings in the Binder/Ashmem implementations.

# KVM Fuzzing

A dedicated effort to improve KVM fuzzing coverage.

- Overhauled KVM API descriptions (Host-side improvements).
- SYZOS Framework (ARM: 2024–2025, x86: 2025):
  - Concept: Instead of random instruction sequences, generate commands interpreted by an immutable guest-side library.
  - Stability: Harder to break by random mutations (parameters mutate, logic stays valid).
  - Capabilities: Unlocks advanced features like IRQ routing and Nested Virtualization.
  - Validation: Allows better, reproducible testing of guest states.
- Running on bare-metal hosts in addition to Google Cloud.

# KFuzzTest: property-based testing for the kernel

Work in progress (under upstream review).

```
FUZZ_TEST_SIMPLE(test_pkcs7_parse_message)
{
    pkcs7_parse_message(data, datalen);
}
```

**/sys/kernel/debug/kfuzztest/test_pkcs7_parse_message/input_simple** is the entry point for the userspace fuzzer(s).

Find out more at KFuzzTest: Targeted Fuzzing of Internal Kernel Functions later today!

2026

# Email Reporting Challenges

We send emails via **GAE**, which does not add proper **DKIM** signatures.

The problem has been highlighted this year, both in the context of **syzbot CI** reports and for normal syzbot reports, e.g. in a recent thread by Linus Torvalds.

We are exploring the ways to address the problem:

- Send emails from a **@syzbot.org** domain, where we can control **DKIM**.
- Send emails from a **syzbot@kernel.org** address.
  - This will likely break our **syzbot+HASH@domain** bug report identification scheme.

# Patch Fuzzing (continued)

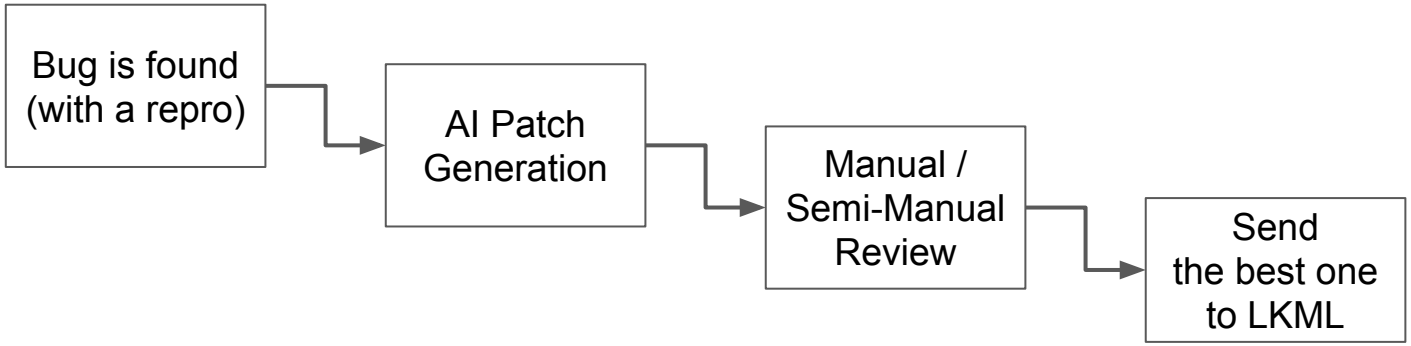
The current system is mostly a PoC, there remains more work to be done:

- Expand to more mailing lists
- Tune the efficiency of the system:
  - Determining the right base tree/commit is a major challenge.
  - Focused fuzzing and finding triage improvements.
- Offer on-demand patch fuzzing to shift it even more left?
  - E.g. by fuzzing all series sent to a special mailing list, before it's sent to the public mailing lists of the corresponding subsystem(s).

# AI Patch Generation?

In 2026, we want to pilot (*) the AI-based patch generation functionality on syzbot.

*(*) High enough precision being a prerequisite.*



# AI Patch Generation: Questions / Discussion

Is it in line with the current AI-generated code policies of the Linux kernel?

What extra information / functionality would simplify the code review?

As a maintainer, what would you demand of such patches?

# AI-based Code Reviews?

Apparently, a hot topic nowadays.

Syzbot CI has the infrastructure to monitor incoming patch series and do processing on them, so it could also take host an AI agent for code reviews.

**But we need to align since similar bots are already deployed by individual kernel subsystems.**

# Questions

Any other questions or comments?

**Also never hesitate to reach out to** syzkaller@googlegroups.com