Linux Plumbers Conference 2022

>> Dublin, Ireland / September 12-14, 2022

Making syzbot reports more developer-friendly

Aleksandr Nogikh
Google

Agenda

- About syzbot (briefly)
- Bug fixing facilitation
- Recent improvements
- Upcoming features
- Discussion

Syzbot – https://syzkaller.appspot.com

Build | Fuzz | Report | Track

Sources
- Upstream stable
- Upstream next
- <...>

Configs
- KASAN
- KMSAN
- KCSAN
- kmemleak
- <...>

Fuzz using syzkaller:

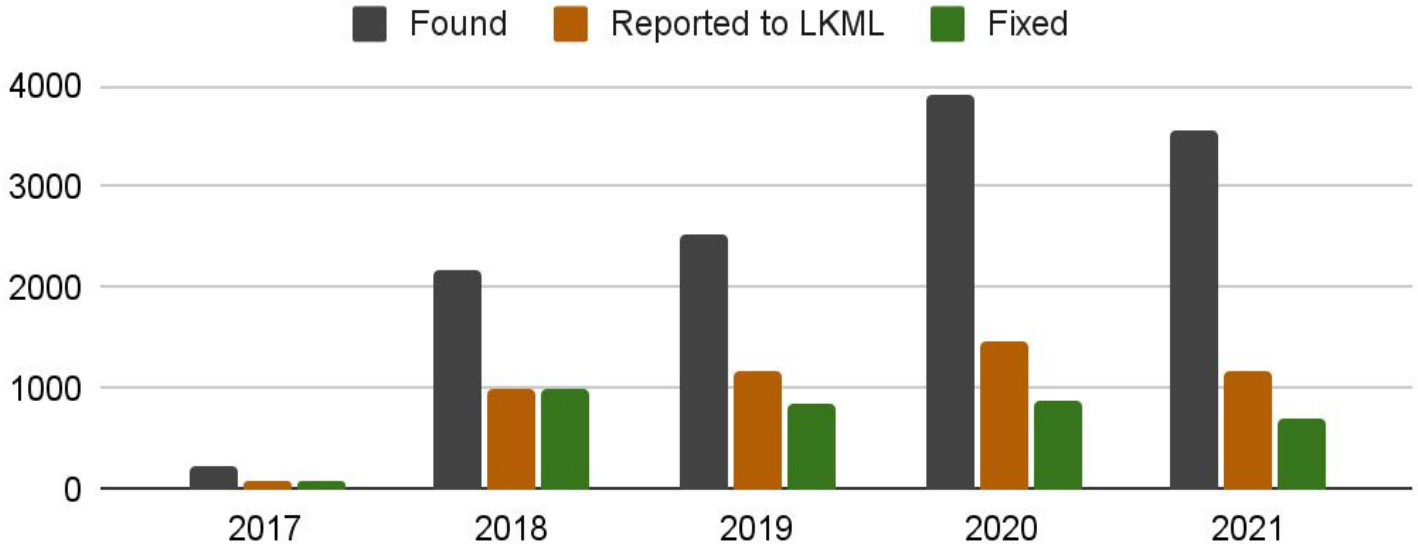
Execute semi-random system calls against booted kernels and capture crashes.

Sanitizers (KASAN, UBSAN, KMSAN, lockdep, etc.) help to detect bugs.

- Symbolize reports
- Extract guilty file
- Invoke get_maintainers.pl
- Send to LKML, maintainers and authors

- Test patches
- Obsolete bugs
- Let users invalidate bugs and mark them fixed

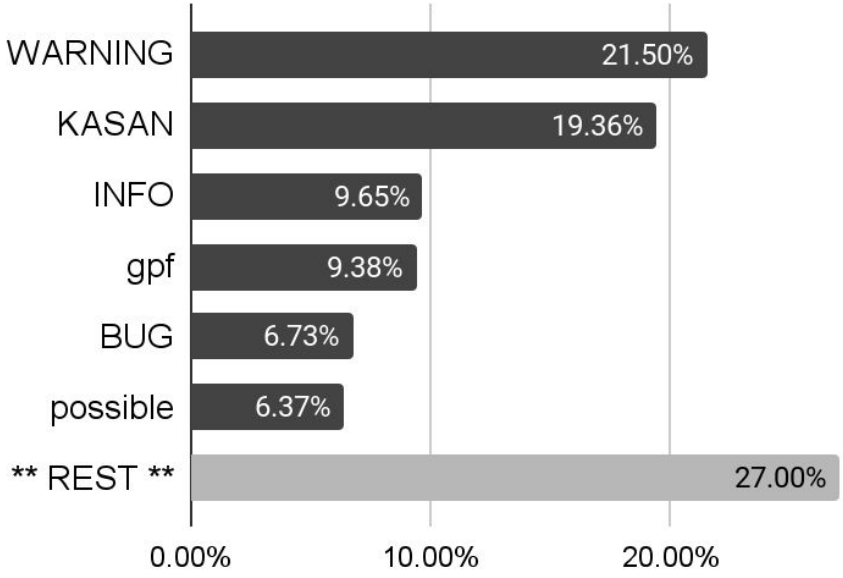
Syzbot stats



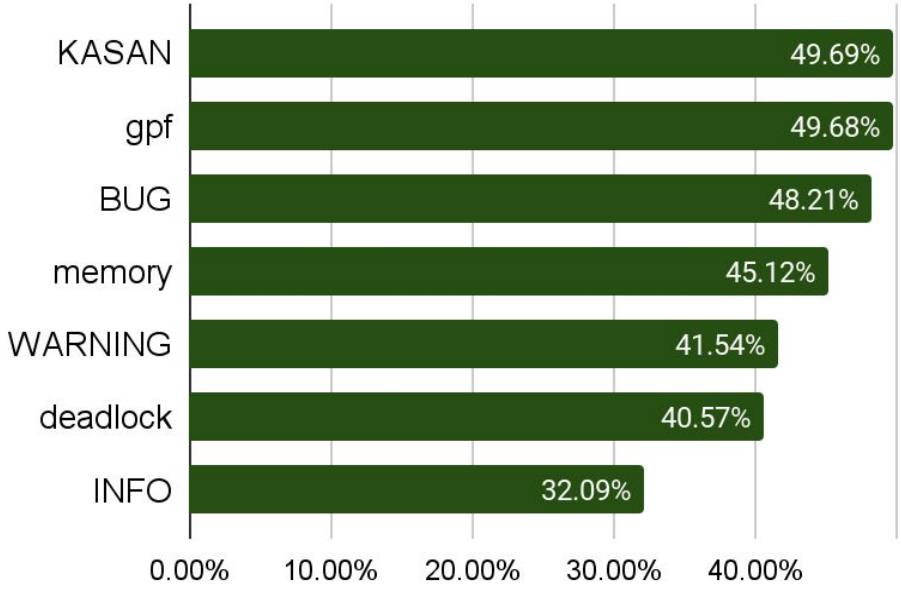
Some bugs do not pass internal pre-moderation and are therefore not reported, some are auto-skipped by syzbot (e.g. single-time rcu stalls without a repro).

Bug types on syzbot

Bug types reported by syzbot since Jan 1, 2020



In 100 days after reporting, what share of bugs will either be fixed or explicitly marked as invalid?



Bug fixing facilitation

Reproducers

Syzbot first attempts to generate a syz repro, then tries to re-reproduce the issue with a C repro.

Syz repro

- Written in a syzkaller DSL.
- Requires an interpreter and an arch-dependent executor running on the target VM.

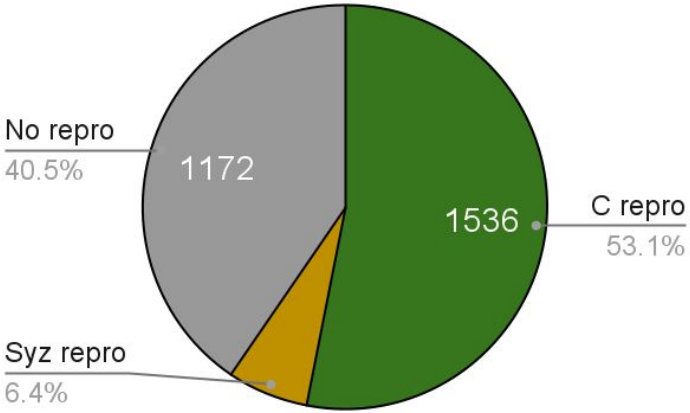
C repro

- A standalone C file that just needs to be compiled and run.
- Usually has a much bigger size.

```
r0 = openat$kvm(0xffffffffffffff9c, &(0x7f0000000000), 0x0, 0x0)
ioctl$KVM_CREATE_VM(r0, 0xae01, 0x0)
r1 = openat$kvm(0xffffffffffffff9c, &(0x7f0000000080), 0x0, 0x0)
ioctl$KVM_CREATE_VM(r1, 0xae01, 0x0) (fail_nth: 37)
```

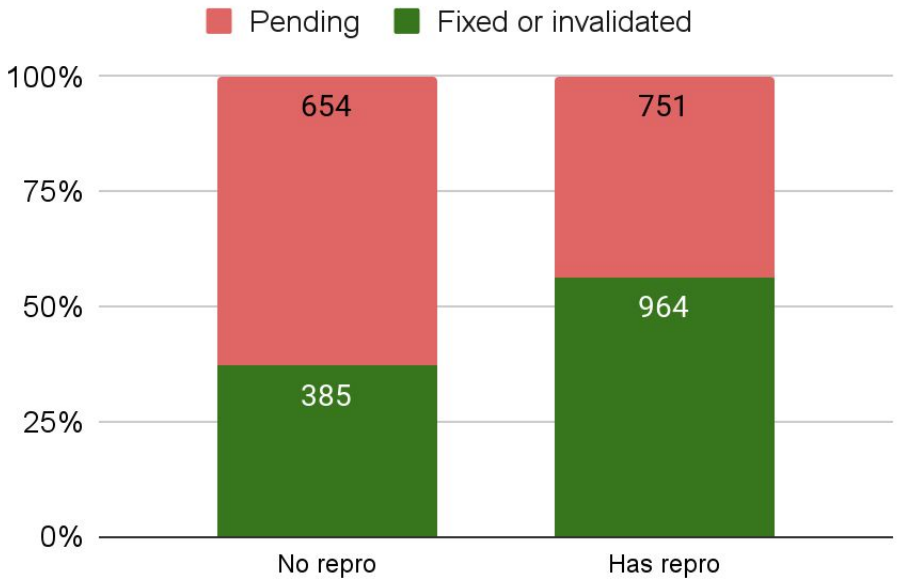
Bug reproduction statistics

Among reported bugs since Jan 1, 2020



Total: 1715

Has a bug been fixed or invalidated 100 days after reporting?



Patch testing

If the reported bug has a repro, it is possible to ask syzbot to apply a patch and run that repro.

One just needs to send a reply to the bug's email thread. E.g.:

```
#syz test: git://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git master

diff -pur a/net/rds/tcp.c b/net/rds/tcp.c
--- a/net/rds/tcp.c     2022-08-21 18:24:08.905058500 +0800
+++ b/net/rds/tcp.c     2022-08-21 18:42:26.529831400 +0800
@@ -166,10 +166,10 @@ void rds_tcp_reset_callbacks(struct sock
```

More details at https://github.com/google/syzkaller/blob/master/docs/syzbot.md

Patch testing: success

@ 2022-08-21 20:43 ` syzbot
  0 siblings, 0 replies; 2+ messages in thread
From: syzbot @ 2022-08-21 20:43 UTC (permalink / raw)
  To: hdanton, linux-kernel, syzkaller-bugs

Hello,

syzbot has tested the proposed patch and the reproducer did not trigger any issue:

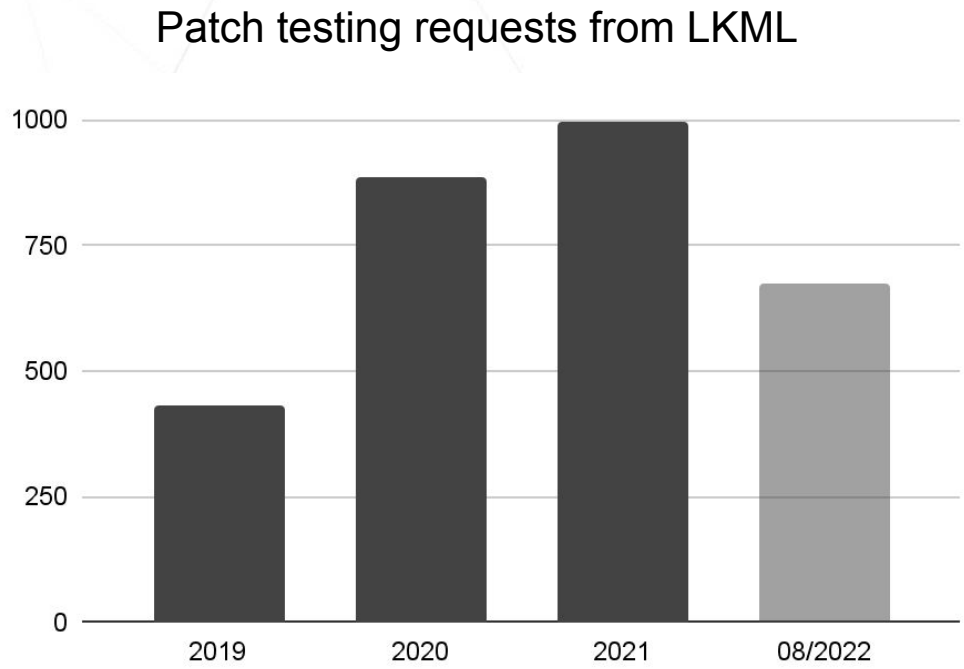
Reported-and-tested-by: syzbot+e696806ef96cdd2d87cd@syzkaller.appspotmail.com

Tested on:

commit: 95d10484 Add linux-next specific files for 20220817
git tree: https://git.kernel.org/pub/scm/linux/kernel/git/next/linux-next.git
console output: https://syzkaller.appspot.com/x/log.txt?x=16cc63d3080000
kernel config: https://syzkaller.appspot.com/x/.config?x=2f5fa747986be53a
dashboard link: https://syzkaller.appspot.com/bug?extid=e696806ef96cdd2d87cd
compiler: gcc (Debian 10.2.1-6) 10.2.1 20210110, GNU ld (GNU Binutils for Debian) 2.35.2
patch: https://syzkaller.appspot.com/x/patch.diff?x=16cac9cb080000

Note: testing is done by a robot and is best-effort only.

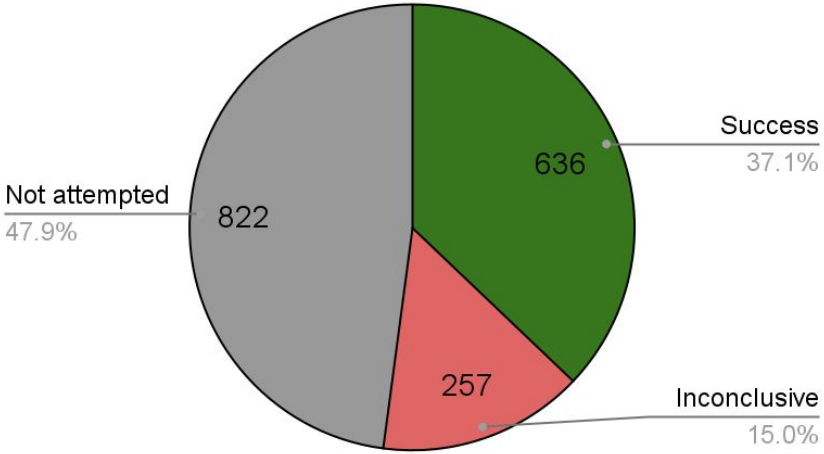
Patch testing: statistics

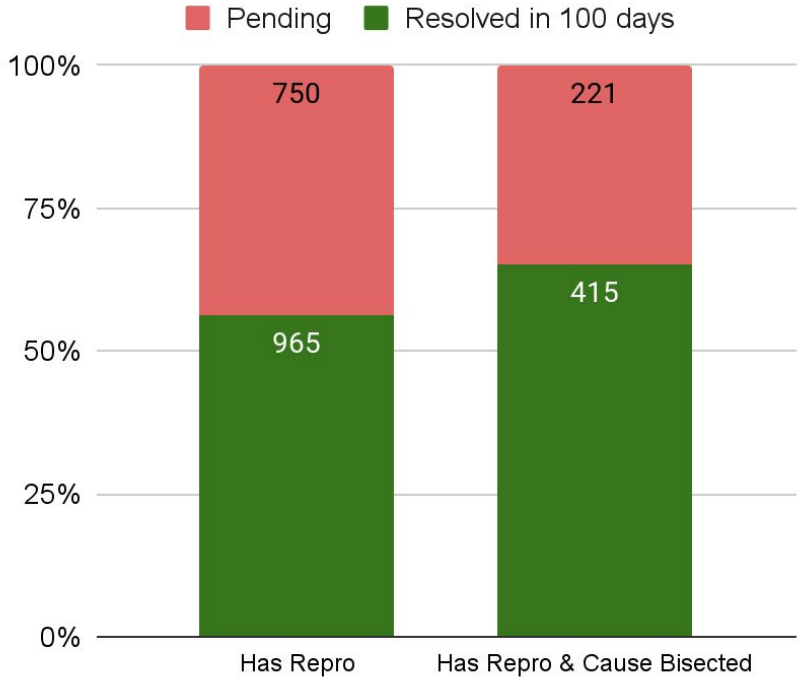


Bisection

For bugs with a repro syzbot performs **cause** and **fix** bisections.

Among reported bugs with repros since 2020.





Recent improvements

Kernel build artifacts

To simplify debugging, we share the following files (starting from September 2022).

- Bootable disk image (works at least for GCE and qemu).
- Kernel object (vmlinux).

syz repro: https://syzkaller.appspot.com/x/repro.syz?x=15065393080000
C reproducer: https://syzkaller.appspot.com/x/repro.c?x=11b22817080000

Downloadable assets:
disk image: https://storage.googleapis.com/syzbot-assets/0cddb4889822/disk-42cf58c2.raw.xz
vmlinux: https://storage.googleapis.com/syzbot-assets/86b24f0bd2f9/vmlinux-42cf58c2.xz

IMPORTANT: if you fix the issue, please add the following tag to the commit:
Reported-by: syzbot+b5d82a651b71cd8a75ab@syzkaller.appspotmail.com

Improvements to bug obsoletion

If a bug was not closed manually (or via fix bisection):

Before: syzbot auto-closes a bug if it's no longer occurring and there's no repro.
Now: syzbot auto-closes a bug if it's no longer occurring and there's no **non-revoked** repro.

Syzbot re-tests each bug reproducer every 100 days.

If it no longer triggers the bug, the reproducer is revoked.

If no repros are left and crashes are no longer happening, the bug is auto-closed.

Since the end of August 2022 there have already been obsoleted > 60 previously opened bugs on our web dashboard.

Strace output

Syzbot tries to run a reproducer under **strace** and captures the output.

strace -e \!wait4,clock_nanosleep,nanosleep -s 100 -x -f <repro executable>

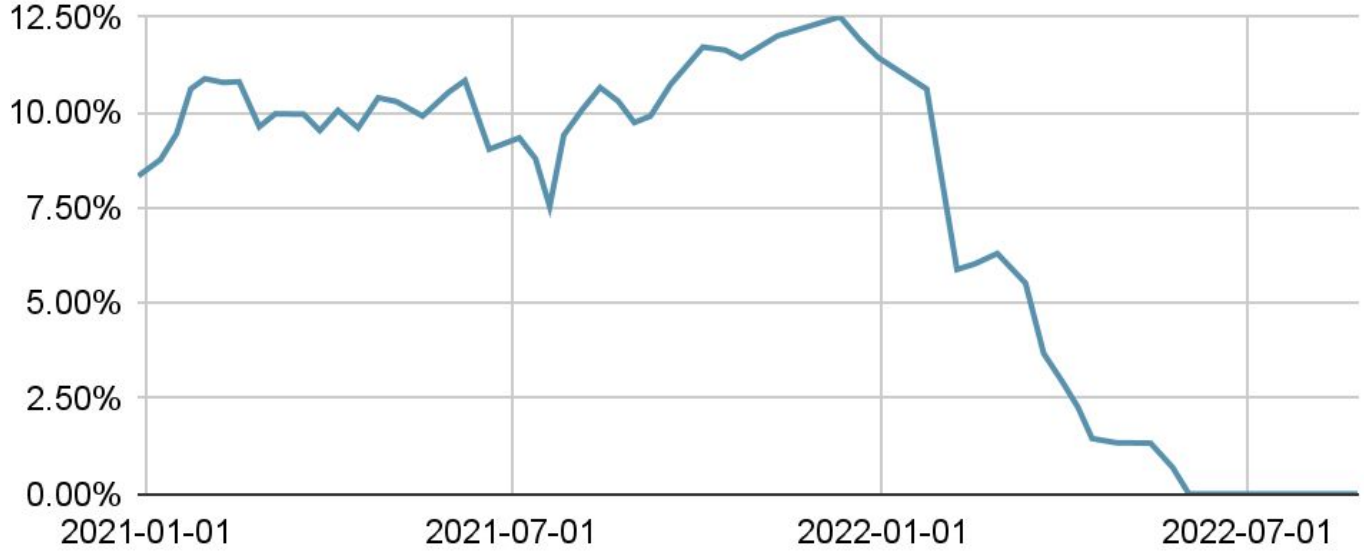
We only capture the output if the execution under strace has led to a crash with same bug title. Currently it is the case for ~60% of runs.

syzbot found the following issue on:

HEAD commit: 7fd22855300e Add linux-next specific files for 20220831
git tree: linux-next
console+strace: https://syzkaller.appspot.com/x/log.txt?x=14e5668b080000

Reduced the negative effect of perf_event_open()

Share of reproducers with perf_event_open() (100 days avg).



Upcoming features

Per-subsystem bug lists

Web dashboard	LKML
Problem Very difficult to filter bugs -- one can only search titles in the browser.	**Problem** Reported bugs get lost -- emails are missed and not revisited by developers, maintainers change over time.
Planned changes Show lists of open bugs for each subsystem.	**Planned changes** Send periodical grouped reminders to individual mailing lists.

Challenge: How to make automatic subsystem detection reliable?

Guilty file extraction

A stack frame is skipped based on the following rules:

Function name:

- memcmp
- memcpy
- show_stack
- ...
- < *many more* >

File name:

- *.h
- lib/*
- mm/kasan/*
- kernel/locking/*
- ...
- < *many more* >

Call Trace:

~~__list_add [inline]~~
~~list_add_tail [inline]~~
~~add_tail [inline]~~
~~klist_add_tail~~
~~device_add~~
hci_register_dev
__vhci_create_device
vhci_create_device [inline]
vhci_open_timeout
process_one_work
worker_thread
kthread
ret_from_fork

hci_register_dev+0x2f6/0xbb0
net/bluetooth/hci_core.c:2593

A subsystem extraction experiment

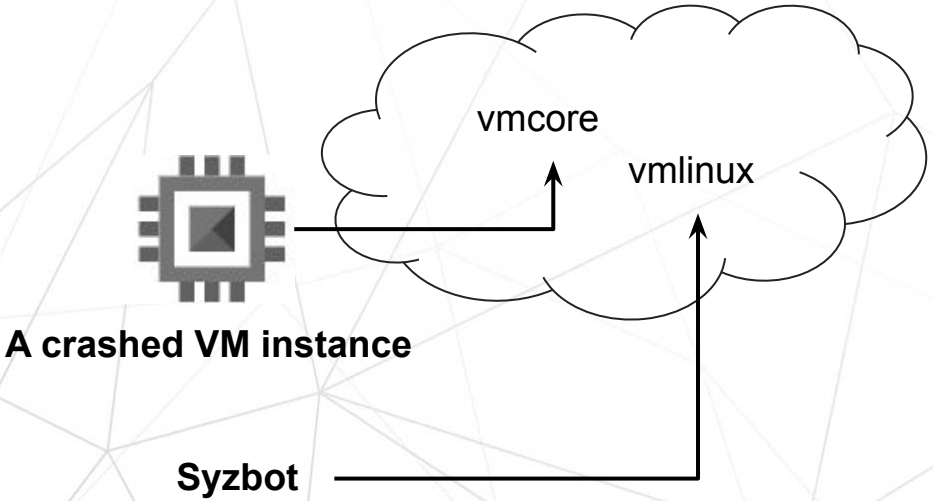
For already fixed bugs, we know their original crash reports and the fixing commits.

Crash reports	**Fixing commits**
Extract the guilty file Invoke **get_maintainer.pl --email -f <file>**	Fetch git patches Invoke **get_maintainer.pl --email <patch>**

Such a straightforward approach can already guess at least one mailing list of the fixing commit in ~75% of cases.

We're investigating ways to further improve that figure.

Per-crash artifacts: kernel core dumps



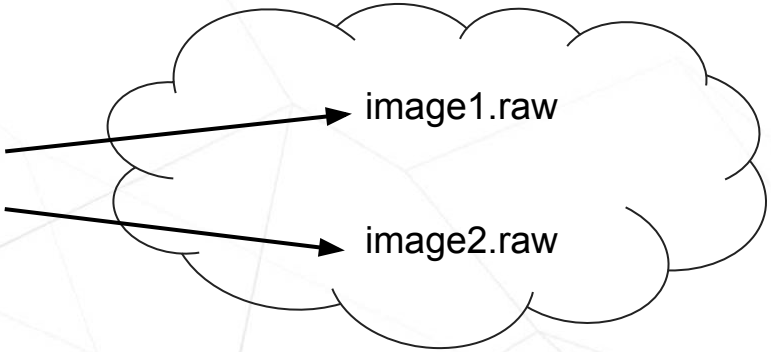
This should facilitate the debugging of crashes for which syzkaller was unable to generate a reproducer.

Per-crash artifacts: mounted images

Reproducer:

syz_mount_image$ext4(< ... >)

syz_mount_image$ext4(< ... >)



The raw images from disk bug reproducers can be difficult to extract manually.

We plan to do this automatically and provide download links.

Thank you for your attention!

If you have any ideas or comments, feel free to share them here at the conference or write us an email:

syzkaller@googlegroups.com

Linux Plumbers Conference 2022

>> Dublin, Ireland / September 12-14, 2022

Making syzbot reports more developer-friendly

Aleksandr Nogikh
Google