

Validate AI bug findings with reproducers

AI finds bugs at scale, but many reports are false positives. A well-argued report is not proof; a kernel crash log is.

— BACKGROUND · AI CLAIM VS. KERNEL EVIDENCE

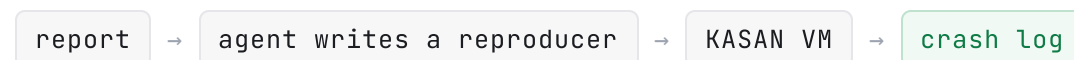
```
AI review → reproducer in QEMU + KASAN

"Potential use-after-free in foo_release(): the refcount
may drop to zero while the object is still queued."
# Plausible. But is it real?
$ ./repro
BUG: KASAN: slab-use-after-free in foo_release
Read of size 8 at addr ffff888012a4cle8 by task repro/214
foo_release+0x1a4/0x2c0
__fput+0x2f1/0xb70
→ verified: REAL · crash log attached
```

— WHY A REPRODUCER IS IMPORTANT

- ◇ Real or not, at a glance. No crash → likely false positive.
- ◇ The exact path. The crash log shows where the kernel breaks.
- ◇ A test for the fix. Rerun the reproducer on the patched kernel.
- ◇ A base for severity. Rerun it with less privilege.

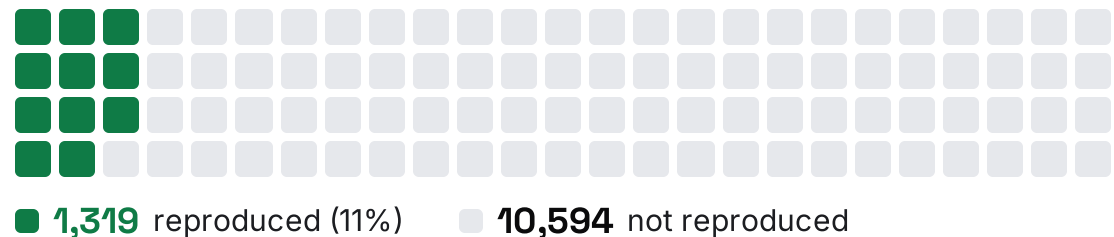
— HOW IT WORKS



— HOW WELL IT WORKS · KERNELCTF



— 11,913 AI BUG REPORTS COLLECTED · 1 SQUARE = 1%



¹ Pu et al. *Patch-to-PoC: A Systematic Study of Agentic LLM Systems for Linux Kernel N-Day Reproduction*, arXiv:2602.07287 (2026).

Finding the bugs to **fix first**

— WHAT WE HIT WHILE FIXING BUGS

Threat model

A real OOB write that needs a modified boot command line. Out of scope.

Odd triggers

Some bugs only fire under strange setups, e.g. a system clock near 1970.

Corner cases need a human

How urgent they are takes an experienced developer to judge.

Review costs too

For some bugs, maintainers find fixing easier than reviewing a patch.

— HOW WE TRIAGE 1,319 REPRODUCED BUGS • FULLY AUTOMATED

STEP 1

Minimum trigger privilege

Get a root reproducer first, then step the privilege down until it stops triggering.

Root

911 • 69%

Namespace

324 • 25%

Unprivileged

84 • 6%

31% of reproduced bugs trigger without root.

STEP 2

Exploitability • the vendor rated it wrong

CVE-2026-43502 • net/rds/message.c

```
bool zcopy = false;
// ...
if (!zcopy)
    __free_page(sg_page(&rm->data.op_sg[i]));
else
    put_page(sg_page(&rm->data.op_sg[i]));
```

An early send failure frees a pinned zero-copy page with __free_page() instead of put_page().

openSUSE advisory

DoS

Primitive we identified

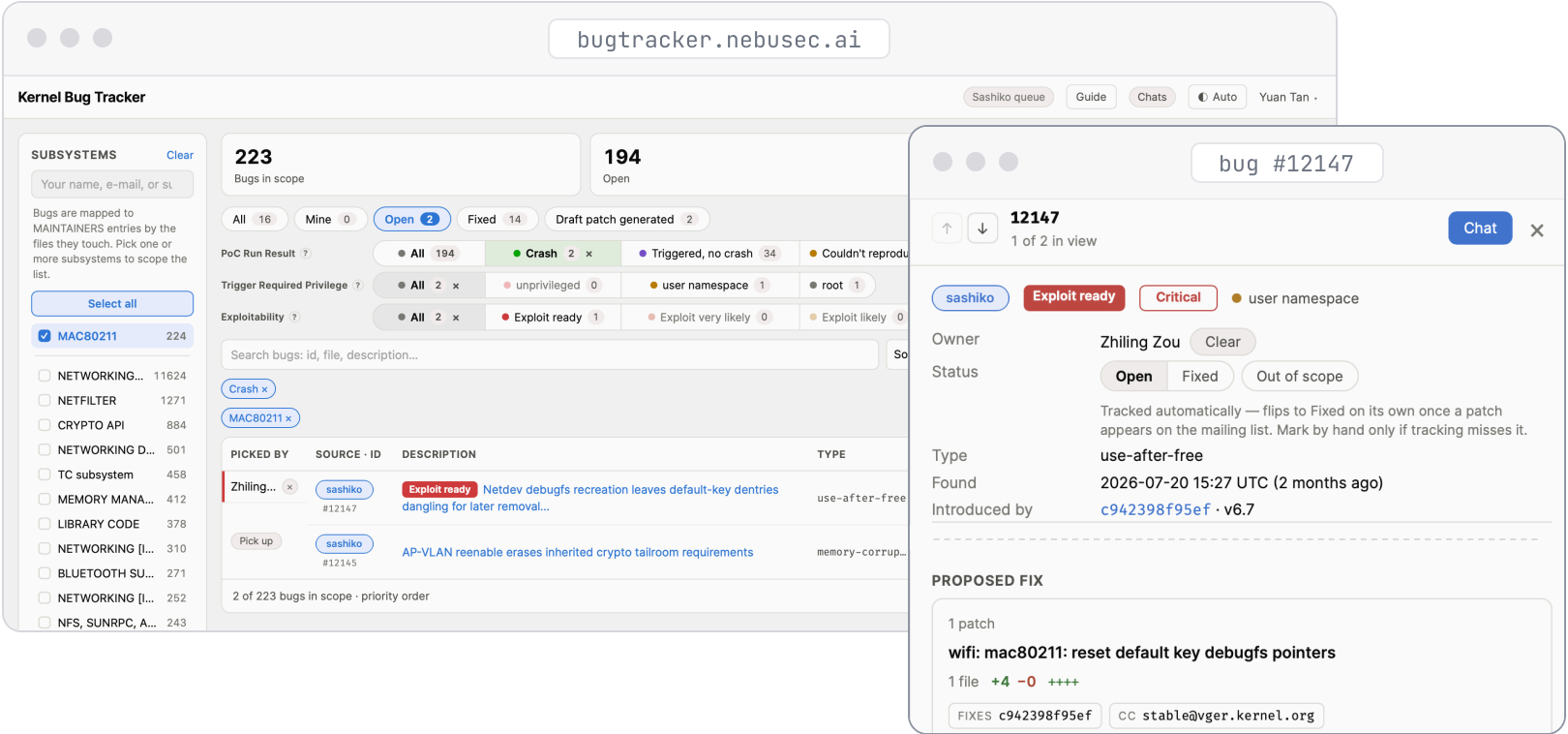
Invalid free

Our working exploit

Local privilege escalation

A platform to collect and validate kernel bugs

Findings from different sources, with reproducers, severity, draft patches and fix status for each bug, in one place.



- Deduplicated findings**
Sashiko, Clashiko and third-party reporters, mapped to subsystems.
- Generated reproducer**
With a crash log (KASAN report or kernel panic) for every bug.
- Severity assessment**
Minimum trigger privilege and exploitability.
- Root cause traced**
The commit and release that introduced it.
- Draft patch**
May be wrong, but a good starting point; fetch it with `b4 am`.
- Fix status tracked**
Marked fixed from lore and git, automatically.