OpenSSL CONFERENCE

Re: [URGENT] Eight Critical 0-Day Vulnerabilities in OpenSSL

Every OpenSSL CVE your distro ships starts as an email to us.

We read it, triage it, fix it, and publish it.

For a decade: ~80 reports a year.

2026: 414 reports by September.

Norbert Pocs · OpenSSL

15.10.2026

Re: [URGENT] Eight Critical 0-Day Vulnerabilities in OpenSSL

Hey team!!

sry would've written a more polite message but I have FOUR deadlines today

ROUND TWO, LET'S GOOOO

I KNOW HOW THIS GOES

I SEND YOU EIGHT (8) CRITICAL ZERO-DAYS

YOU SEND ME ONE (1) LINE ASKING FOR A REPRODUCER BC YOU THINK I'M A BOT

I SEND YOU A REPRODUCER

YOU SAY "OUT OF THREAT MODEL"

I SAY YOUR THREAT MODEL IS OUT OF MY THREAT MODEL

(jk, love you all, you're doing great. mostly.)

So my agents (all 40 of them, they have names now, it's getting weird)
spent the weekend chewing through the codebase and flagged EIGHT findings.
I then triaged every single one PERSONALLY with my own human eyes, which,
for the record, are VERY experienced eyes.

1. heap overflow (CRITICAL)

ossl_ticket_frobnicate() trusts the length byte. ASAN agrees with me:

```
==31337==ERROR: AddressSanitizer: heap-buffer-overflow on address
0x602000000042 at pc 0x55d00dfeed42 bp 0x7ffc0ffee000 sp 0x7ffc0ffedff0
READ of size 4096 at 0x602000000042 thread T0
    #0 0x55d00dfeed41 in __asan_memcpy
    #1 0x55d00dfeed99 in ossl_ticket_frobnicate crypto/frob/frob_lib.c:42:9
    #2 0x55d00dfeedaa in ossl_ticket_unfrobnicate crypto/frob/frob_lib.c:87:12
    #3 0x55d00dfeedbb in main test/frob_test.c:13:5
```

2. the same overflow, but on Tuesdays (CRITICAL)

3. timing side channel (CRITICAL, if you squint)

4. memory leak (CRITICAL*)

*if you run it two billion times in a loop

5. the documentation is wrong (CRITICAL for my mental health)

6-8. see attached PDF (63 pages, executive summary starts on page 41)

fix

patch attached. it compiles. (on my machine.)

OK rant time

your ticket frobnication code is a house on fire, and you're all sitting
in the living room admiring the warm glow. years and years of tech debt,
and now the bill is here, and it's gonna get a LOT worse before it gets
better.

Do you want help, or do you prefer the ostrich approach, which seems to be
working SO WELL FOR YOU? As we say back home, what you're doing right now
"is like bailing out a boat with a fork". no time to explain, go look it up.

My cleanup PR, which is LITERALLY MOVING CODE FROM ONE FILE TO ANOTHER,
has been open for 23 days.

AWESOME. GREAT JOB. I CAN TELL YOU'RE TAKING THIS SO SERIOUSLY.

FYI: my agents are currently finding > 1 bug PER DAY in this code.

Let me write that again, it's not a typo: > 1 BUG PER DAY

YES THAT IS A GREATER THAN SYMBOL

and it takes > 3 WEEKS to merge a copy/paste PR

YES THAT IS ANOTHER GREATER THAN SYMBOL

I'm not a mathematician, but I'm sure one of you is reading this: how does
that math work out?

BECAUSE WITH MY STICKS-AND-PEBBLES MATH, IT DOES NOT LOOK GOOD

If this level of engagement continues, I might just start live streaming
these to the mailing lists, no embargo. your call. it's your repo not mine.

NOW LET'S HAVE A SUPER PRODUCTIVE CONVERSATION

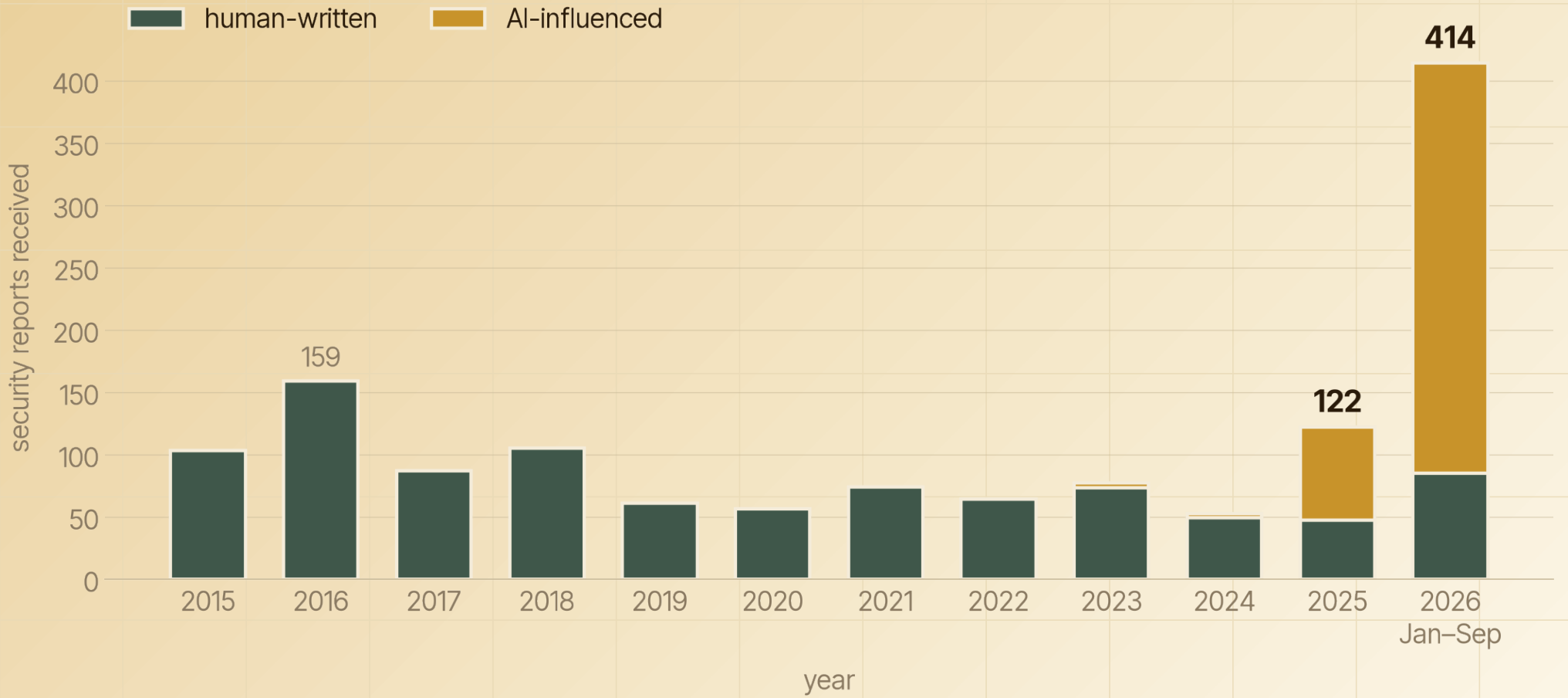
Cheers

The issue is not the increased amount of reports.

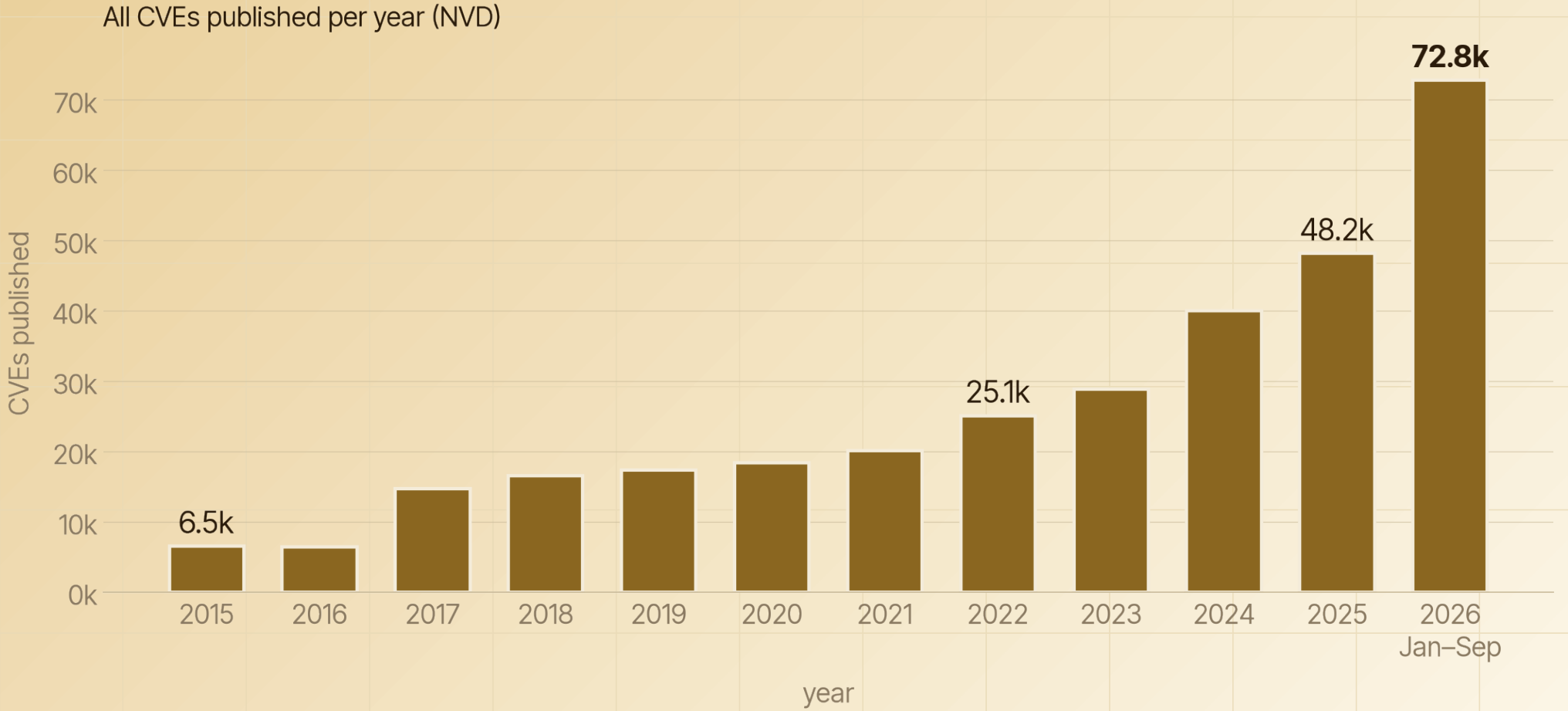
The problem is how we handle them.

Human reports stayed flat. The rest is new.

OpenSSL CONFERENCE



Not just us


All CVEs published per year (NVD)
CVEs published
70k
60k
50k
40k
30k
20k
10k
0k
6.5k
25.1k
48.2k
72.8k
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
Jan–Sep
year

Three places it hurts

1

Intake

Processing what comes in

2

Internal process

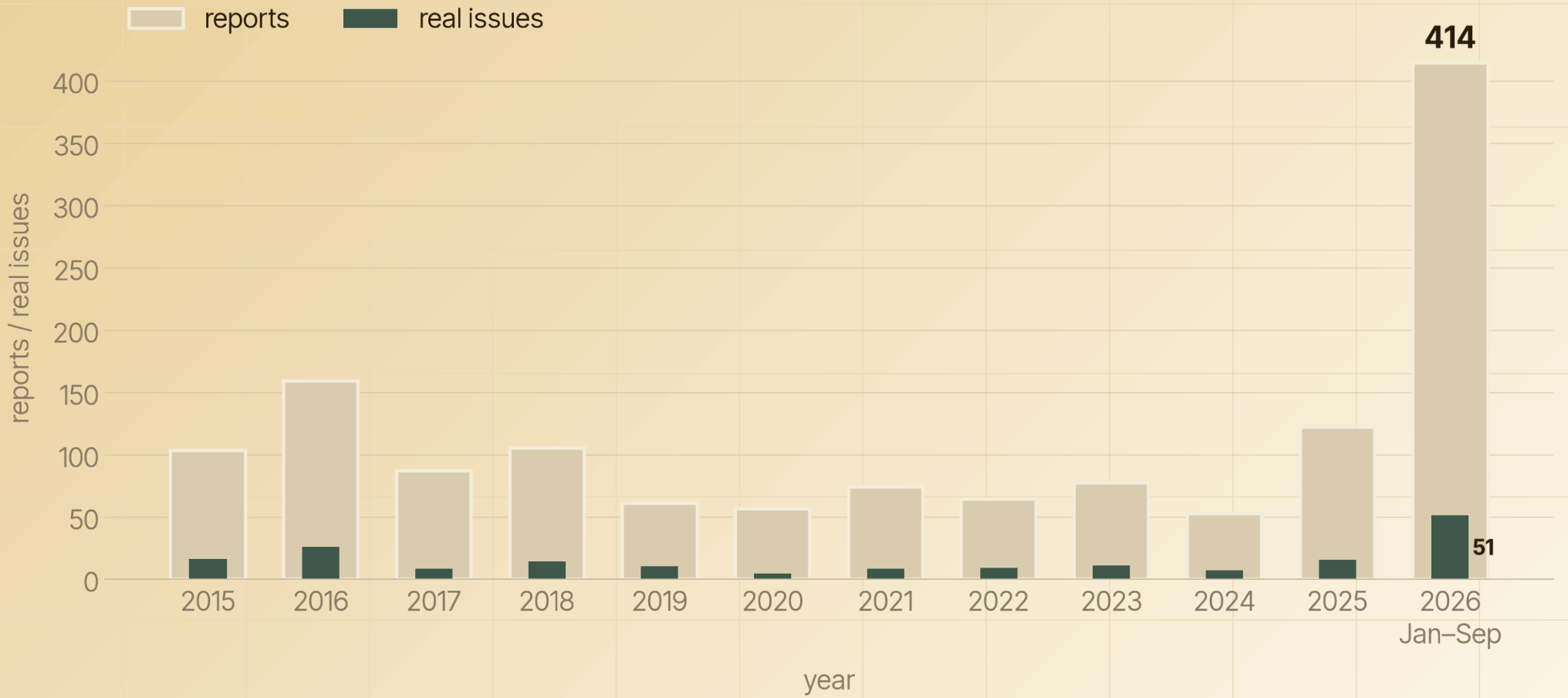
Work starts late, ends in a rush

3

Release

Most CVEs wait for a release

More reports, not more issues



Every report takes longer to read

1,400 → 3,100

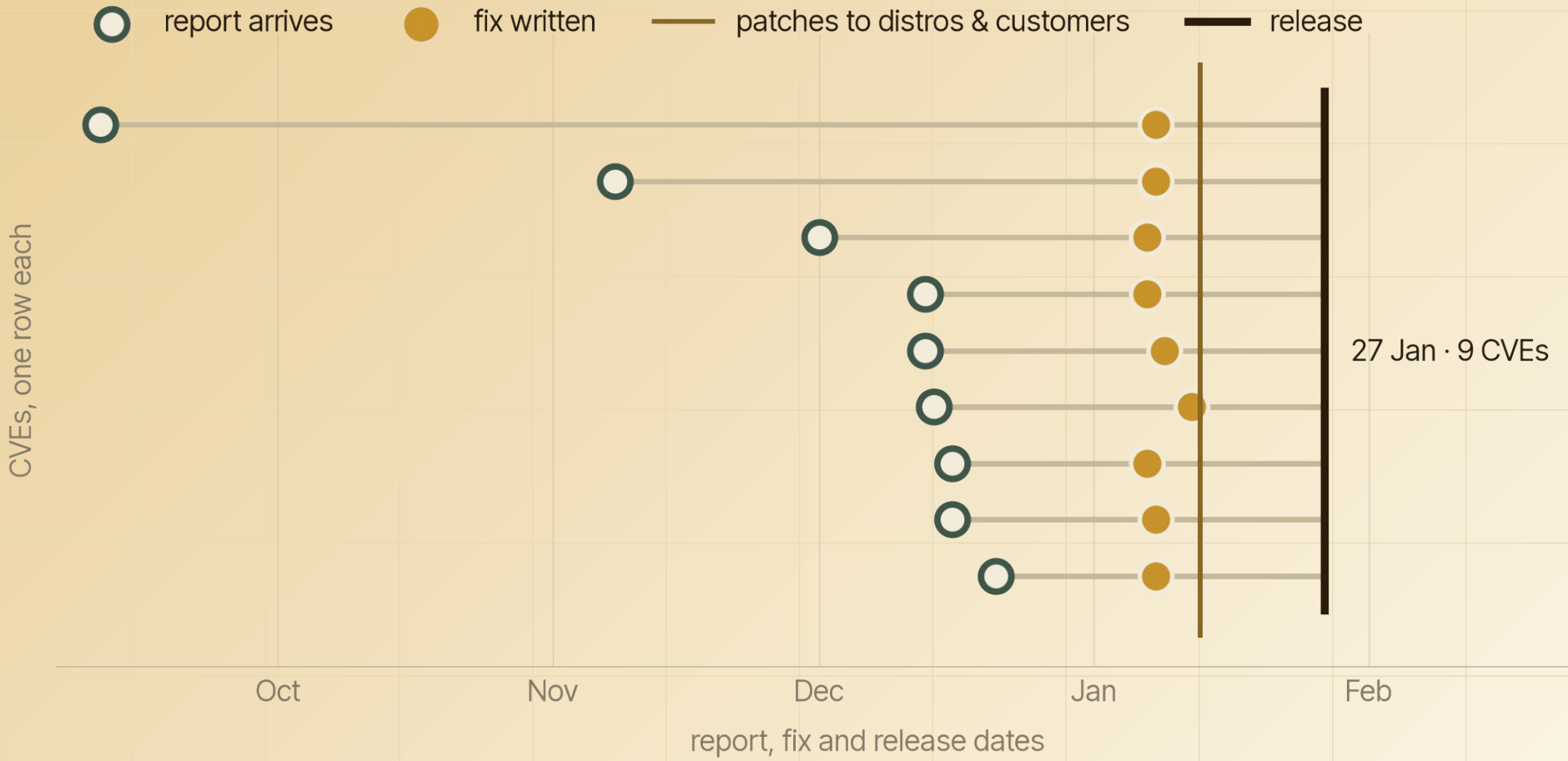
characters: median length of a report, before 2023 vs. 2025–26

From block to stream

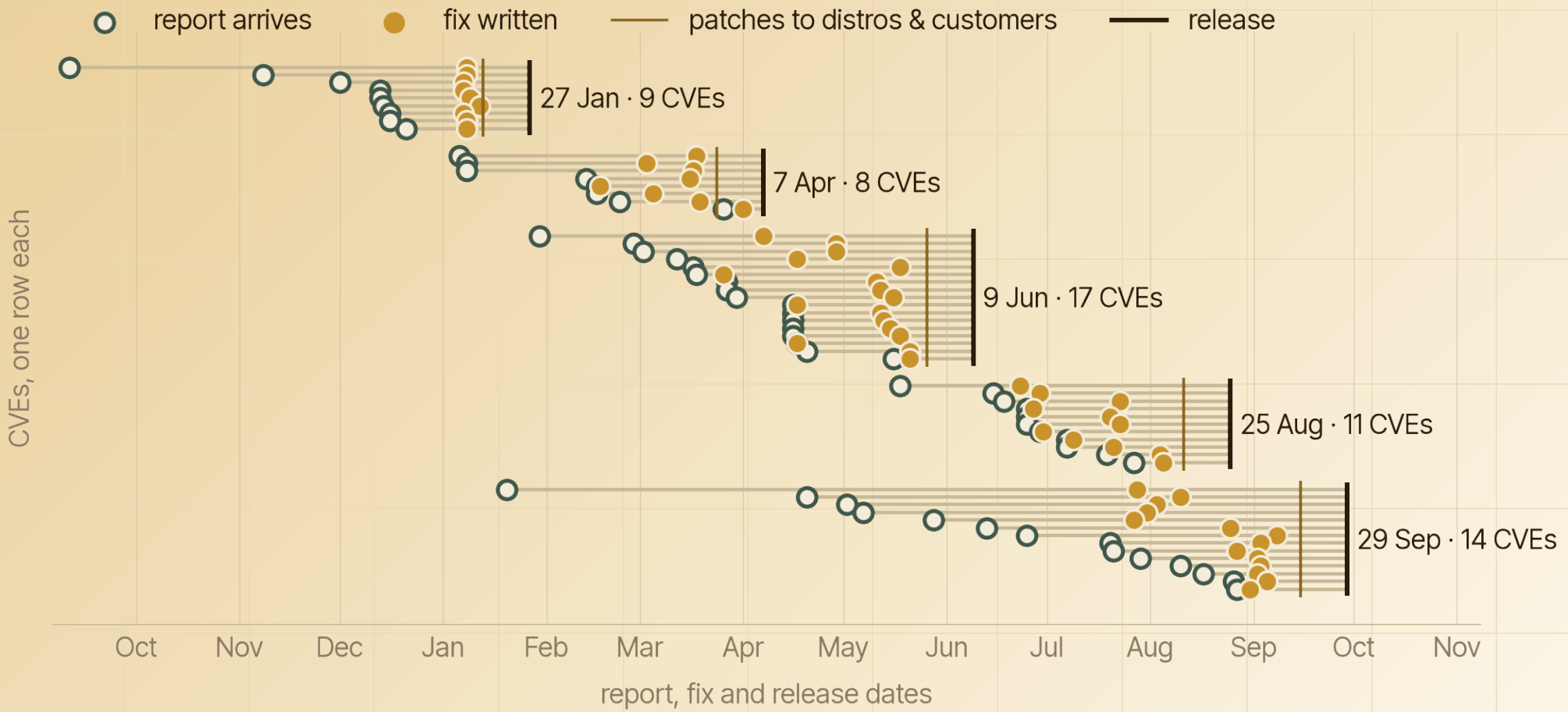
Remove the parts where we wait.

- **Block cipher**
- Collect a full block, then process it.
- **Stream cipher**
- Process each byte as it arrives.
- **Today**
- Collect reports until a release, then fix them all.
- **Goal**
- Fix each report as it arrives.

One release, up close



Every release, all year



What block mode costs

- **Block (today)**
- Fix work starts late
- Fix PR and advisory PR come late
- Many review rounds
- Last-minute rush blocks other development

- **Stream (goal)**
- Work on issues as they arrive
- Workload spread over the lifecycle
- Release when ready

Where does your pipeline wait?

Fix Lows in public

A direction we are considering.

- **Fix it in public, as it arrives**

  No embargo, no waiting for a release

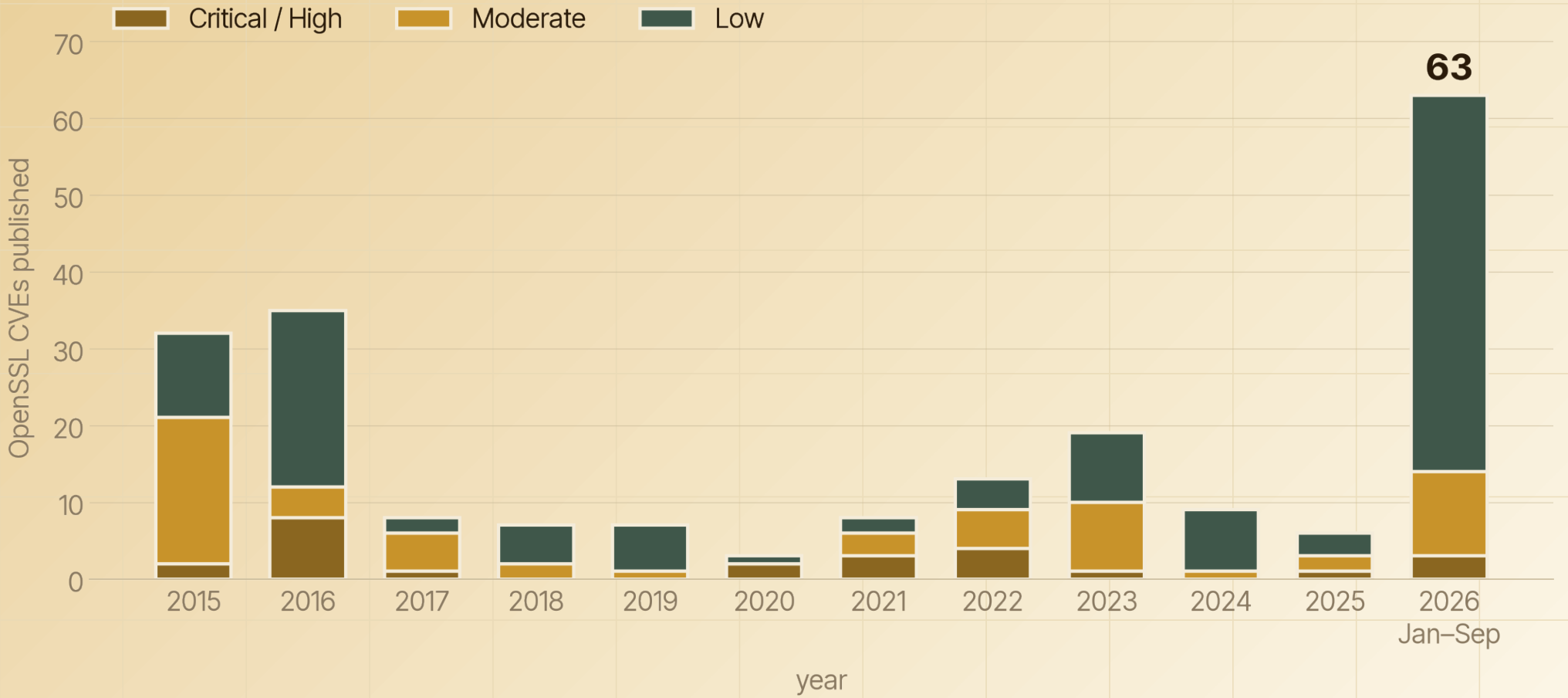
- **Announce it**

  Advisory and CVE, as today

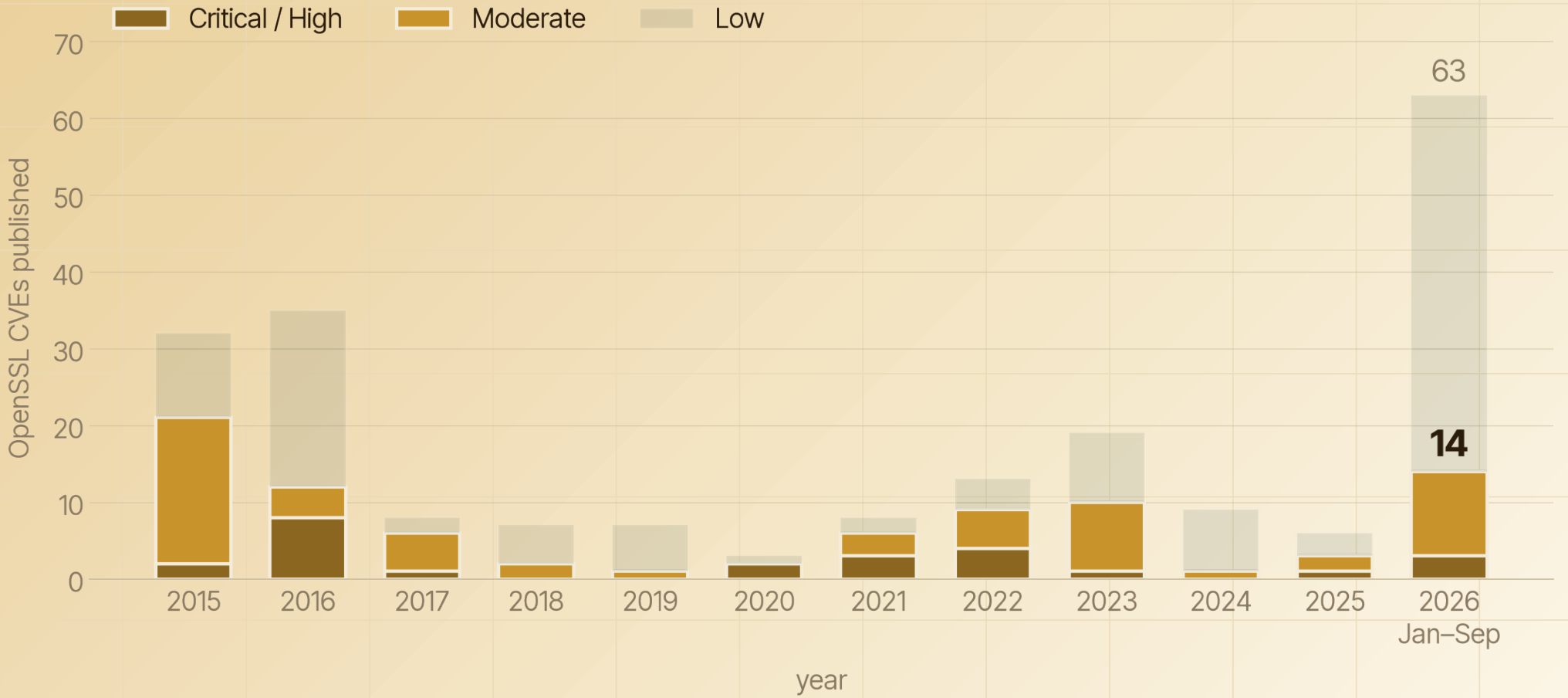
- **Ship it in the next release**

  Nothing changes for the release itself

Take the Lows out



Take the Lows out

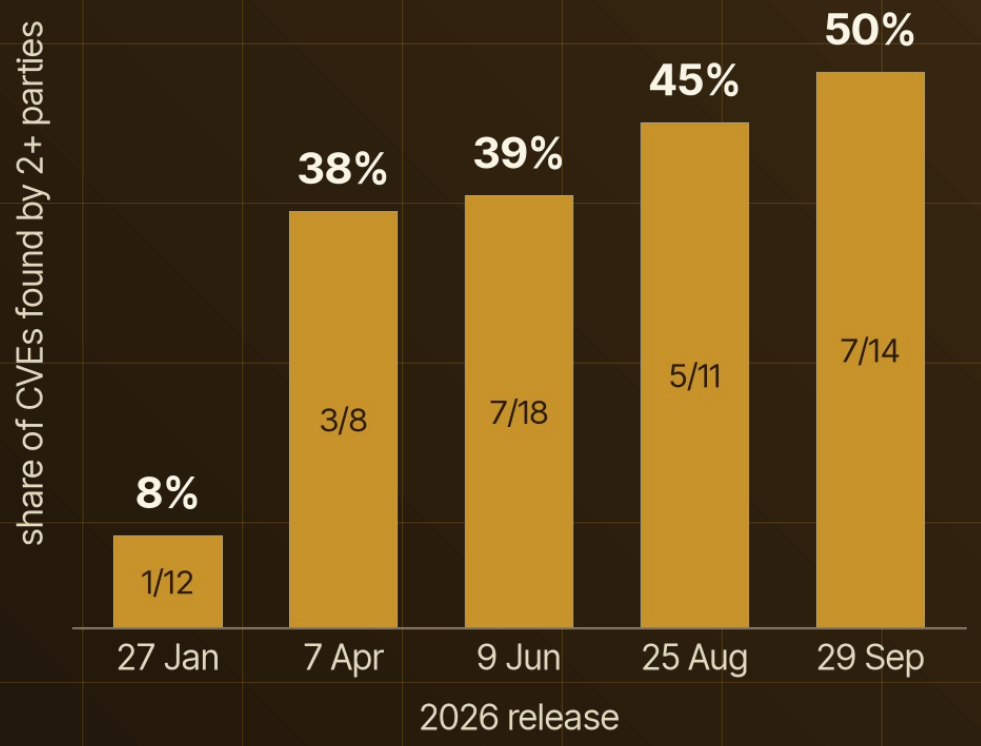


6

independent parties reported the same Low

CVE-2026-28389

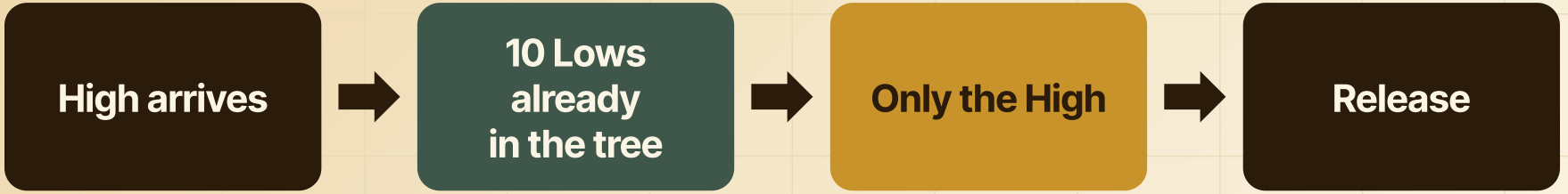
CVEs found by more than one party, per 2026 release



Ready for the next High

Today

Lows streamed



The answers

- **How does AI change security handling?**
- Reports grow faster than the team.
- The process has to change.

- **What is the shortest release cycle?**
- Stream the Lows.
- Release the severe ones fast.

Thank you!

Questions?

Norbert Pocs · OpenSSL