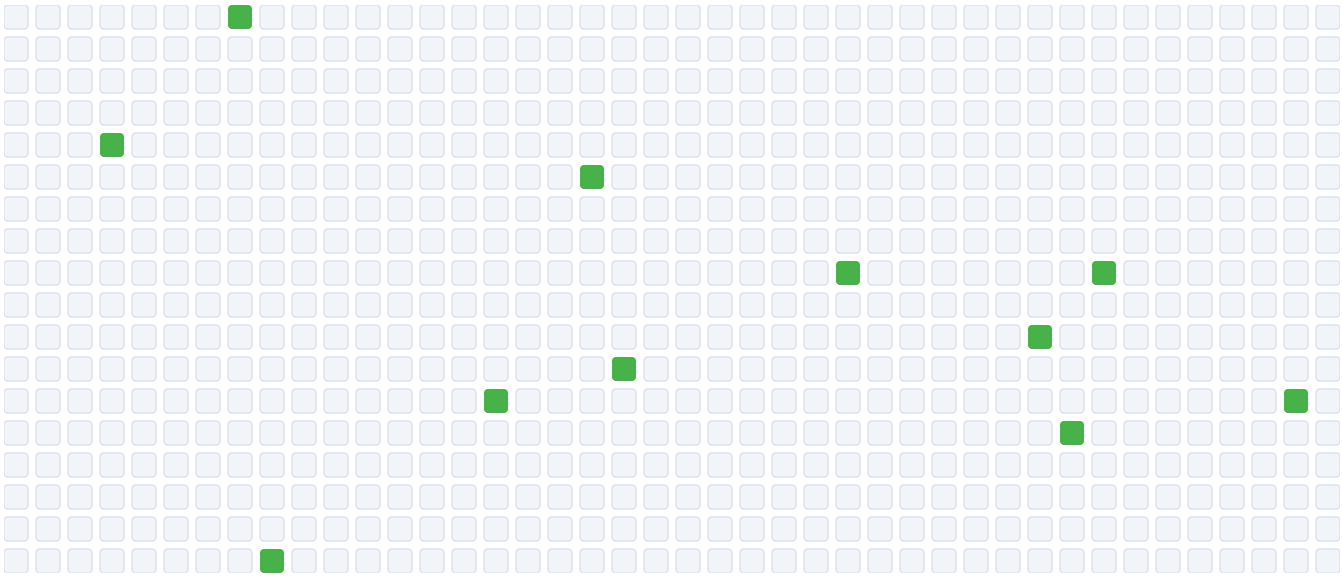
evidence

The Vulnpocalypse Report

Every confirmed in-the-wild exploitation of a published CVE, January 2018 through July 2026, measured against the forecast that AI would change everything.



253,912 CVEs published. Adversaries exploited 1.5%.

The Vulnpocalypse Report

The security industry keeps predicting that AI will supercharge vulnerability exploitation. We went looking for the evidence. We traced every confirmed in-the-wild exploitation of a published CVE from January 1, 2018 through July 15, 2026 (all 3,769 of them) and measured how much adversary behavior has actually changed.

253,912 CVES PUBLISHED FROM JAN 1, 2018 THROUGH JUL 15, 2026 (8+ YEARS)

116 days
Median gap between patch release and first exploitation, for CVEs from Jan 1 to Jul 15 in 2026.

+11.5 days
Average annual increase in the gap since 2018, from 24 days to 116. Defenders are gaining time.

~1.5%
Of 253,912 published CVEs, adversaries exploited 3,769 in the wild.

0.28%
Adversaries struck 711 CVEs before a patch existed. Those are the true zero-days.

1 in 5
Exploited CVEs each year is a zero-day, a ratio that has held flat for eight years.

About Root Evidence

Root Evidence builds evidence-based vulnerability management. We identify the CVEs that have historically led to financial loss, so security teams can put their hours where the losses happened and report cyber risk in dollars.

Contents

Executive Summary

For the past two years the security industry has been predicting that AI will make the vulnerability problem catastrophically worse: far more vulnerabilities, weaponized within hours by people with no real skill. The forecasts run from alarming to apocalyptic, and pundits have even named the moment.

They call it the Vulnpocalypse.

We went to the record to check. We traced every CVE with confirmed in-the-wild exploitation from January 1, 2018 through July 15, 2026 and measured what adversaries actually did as AI arrived.

TL;DR: They barely changed.

AI is filling the CVE catalog at record speed, and adversaries are still working the same narrow slice of it they always have. Security people routinely confuse what's possible with what's probable, and the Vulnpocalypse hype exploits that confusion.

Of the 253,912 CVEs published in that window, adversaries exploited 3,769 in the wild, which is 1.48%. They have never touched the other 98.5% in any recorded attack. They exploited just 711 CVEs, 0.28% of everything published, before a patch existed. Whatever picture you carry of adversaries routinely chaining zero-days to break in, the breach record and the financial loss record show something different. Most software that can be hacked never is.

The rest of the CVEs are n-days, exploited after a patch was already available. For the overwhelming majority of n-days, the oft-repeated claim that time-to-exploit has collapsed to hours is simply false. Adversaries who exploited a CVE in 2026 waited a median of 116 days after the vendor patched. In 2018 they waited 24. They're patient because they can be: researchers keep publishing bugs faster than defenders remediate them, so an adversary gains nothing by racing when the same unpatched target will still be standing in month sixty.

AI has changed one group's behavior so far, and it's the people *finding* bugs, not the people *using* them. Researchers and vendors published 15% more CVEs in 2025 than in 2024, and they're on pace for nearly a third higher in 2026. Something is speeding them up, and AI is clearly part of it, though we can't yet isolate how much.

What we *can* measure is the ratio of what gets found to what gets exploited: AI-assisted researchers now find vulnerabilities far faster than adversaries choose to use them. Adversaries exploited a flat-to-falling share of each year's published CVEs, and since 2022 zero-days have held flat at 15% to 20%, with 2026 tracking at the low end.

That's not the Vulnpocalypse we were promised.

If researchers keep finding bugs faster than adversaries weaponize them, the exploited percentage keeps falling, and each new year of CVEs will bury the same short list of dangerous bugs under more noise. A defender who counts CVEs will find every year scarier than the last, even as adversaries keep digging into an ever-smaller slice of the pie.

Defenders should take their priorities from what adversaries do, instead of focusing on what researchers count. Patch what adversaries use. Know whether you're facing a zero-day or an n-day, because you defend against each differently. CVE count is one of the loudest measures, but it's also one of the least valuable.

Check Thevulnpocalypse.com: 147 days into the apocalypse, and civilization is holding up fine. Maybe the Vulnpocalypse arrives someday. We wrote this report to answer a more useful question, which is what adversaries are doing right now, by the numbers. We would rather be accurate than alarming.

Jeremiah Grossman, CEO, Root Evidence

Why Our Numbers Are So Different (and Whose to Believe)

If you've seen a chart this year showing exploit windows collapsing toward zero, it almost certainly came from ZeroDayClock. Sergej Epp, former CISO at Sysdig, built the site, and it tracks what it calls Time-to-Exploit across 83,000+ CVEs drawn from ten sources, including CISA KEV, ExploitDB, and Metasploit.

Its headline findings run from dramatic to dire: mean time from disclosure to exploitation down from nearly a year in 2021 to about a day now, one hour projected by 2027, one minute by 2028, and most exploited vulnerabilities now counted as zero-days. Major tech and security companies have signed on, journalists have covered the charts widely, and vendors already quote its numbers in their pitch decks.

When someone tells you defenders have hours to patch, they're usually quoting ZeroDayClock, whether they know it or not.

We checked its math, and we're spending a section on it because the errors we found produce most of the Vulnpocalypse's headline numbers.

ZeroDayClock (ZDC) counts an exploited CVE as a "zero-day" if adversaries used it on or before the day the National Vulnerability Database (NVD) published its record. NVD is the US government's vulnerability catalog, run by NIST. It indexes and enriches CVE records after publication, and lately it has averaged 25 days behind the vendor's patch.

Figure 1. Zero-day vs. n-day: the core distinction

Defenders' options depend on which one they face.

Zero-Day	N-Day
Adversaries exploited the CVE before any patch existed. The vendor had not shipped a fix, so defenders could do nothing but wait for one. We analyze the dataset's 711 true zero-days in the separate section *True Zero-Days: What the Data Actually Shows*.	Adversaries exploited the CVE after a patch existed. N counts the days between patch release and first exploitation: the window defenders had to close the exposure. Whether they patched inside that window decided the outcome. We analyze the 3,058 n-days in the separate section *N-Days: The Defender Window*.

So under ZDC's method, a vulnerability an adversary exploited three days after the patch ships still counts as a zero-day, because NVD hadn't caught up yet. But defenders had access to a working patch for those three days.

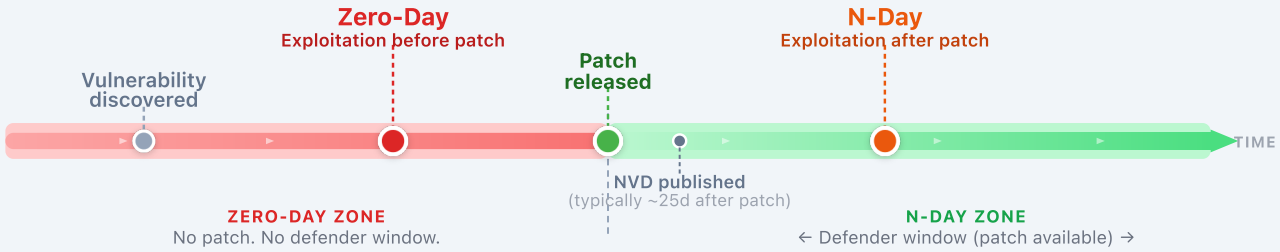
Our take: Nobody should call that a zero-day.

Historically, the cybersecurity industry never defined a zero-day this way before. Practitioners have always measured time-to-exploitation from the patch to the exploit, because that's when the defenders have the ability to react. When ZDC starts the clock at NVD publication instead, it credits NVD's backlog to the adversary, and its numbers stop describing what happens in the wild. We don't know whether ZDC chose this definition deliberately or inherited it from AI slop. Either way, we reject the findings it produces, and this report will show why.

We measure from the patch to the exploit, the right way. For every exploited CVE in the dataset, we checked whether a fix existed at the moment exploitation was first confirmed. Where no fix existed, we classified the CVE as a zero-day: the vendor hadn't shipped anything defenders could apply. Where a patch was already available, we classified it as an n-day, where N counts the days defenders had to patch before the first exploit hit.

Security teams know there's a difference between a true zero-day (where no patching cadence would have changed the outcome) and an exploit for an n-day vuln (where a fix existed and timing mattered). By starting the clock at patch availability, we aim to answer the question security teams actually face: was there a patch, and how much time did we have?

FIGURE 2. ZERO-DAY VS. N-DAY: WHERE EXPLOITATION FALLS ON THE TIMELINE. SCHEMATIC, NOT TO SCALE.



Where Our Data Comes From

Sources: CISA Known Exploited Vulnerabilities (KEV) and VulnCheck KEV, merged on the earliest confirmed exploitation date per CVE. Each CVE represents a vulnerability with confirmed active exploitation in real-world attacks. The full dataset holds 3,769 CVEs spanning January 1, 2018 through July 15, 2026. The True Zero-Days section analyzes the 711 true zero-days, the CVEs where confirmed exploitation occurred before any patch was available. The other 3,058 CVEs (exploited after a patch existed) are analyzed in the N-Days: The Defender Window section. Two further confirmed-exploited CVEs have no patch date and are excluded from these counts. Classification methodology and patch date sources are detailed in the Data Methodology.

Patch dates were collected from vendor advisories, Microsoft Security Response Center (MSRC), the MITRE CVE API, and primary sources including vendor changelogs, GitHub releases, and security advisory pages. For every CVE, the earliest date a patch was publicly available is what the analysis uses. This matters because most CVE databases record when they processed the advisory, not when the patch shipped.

The full data pipeline, classification logic, and edge-case handling are documented in the Data Methodology. Every figure in these reports traces back to those rules.

For every CVE, the analysis records when a patch became available, where that date came from, whether exploitation came before or after it, and how long defenders had. Every case where an estimated patch date was later replaced with a verified one is documented with its source and reason.

Key Findings

1. **Is AI accelerating vulnerability discovery? Likely yes.**

   Researchers published 15% more CVEs in 2025 than in 2024, and they're on pace to go nearly a third higher again in 2026. Something is driving the surge, and AI is almost certainly contributing. We can't yet pin the increase on any single cause, so the full extent of AI's influence is TBD.

2. **Is AI increasing the number of exploited vulnerabilities? No.**

   If anything, adversaries are exploiting a slightly *declining* share of what researchers publish. Of the 253,912 CVEs published since 2018, adversaries have exploited 3,769 in the wild (1.48%) and have never touched the other 98.5% in any recorded attack. Researchers keep widening the pool, but adversaries keep fishing in the same corner of it.

3. **Is AI driving more zero-days? No.**

   Since 2022, adversaries have struck before a patch existed on a stable 16% to 21% of the CVEs they exploit, with no sustained upward trend, and in 2026 they're tracking closer to 15%. Across the full eight years, adversaries managed pre-patch, true zero-day exploitation on just 711 CVEs, 0.28% of the total published.

4. **Is AI accelerating the speed at which adversaries exploit vulnerabilities? No.**

   Adversaries who first exploited a CVE in 2026 waited a median of 116 days after the vendor patched it. In 2018 they waited 24. As researchers publish more and defenders patch less of it, adversaries gain nothing by rushing. They can keep working the growing backlog of unremediated bugs at their leisure, regardless of patch availability.

5. **Are adversaries exploiting at least some vulnerabilities fast? Yes.**

   The median hides a split population. Adversaries hit roughly a third of 2026's n-days within 30 days of patch release, and they took more than a year to get to another third. No single patching SLA covers both groups, and the fast-moving third is where a 30-day target matters most.

6. **What do the exploited vulnerabilities look like? They cluster.**

   For zero-days in commercial software and mobile platforms, adversaries lean on memory corruption and injection/RCE. For n-days, they favor authentication bypass and injection. The 208 WordPress plugin zero-days follow the same web-application pattern: overwhelmingly injection and access-control flaws. And adversaries return to the same short list of software vendors year after year.

How Fast Adversaries are Exploiting Vulnerabilities

When a vendor ships a patch before adversaries find the flaw (and that's the order of events for four out of five exploited CVEs) defenders get a window of time to apply it. Figure 3 measures that window with two different clocks. ZeroDayClock starts its clock when NVD publishes the CVE record, producing the amber line, its Time-to-Exploit (TTE). We start ours when the vendor releases the patch, because that's the first moment a defender can act. The green line, what we call the defender window, shows the median days adversaries then waited before striking.

116d
median days adversaries waited after the patch before striking, 2026 n-days through mid-July.

327 days
peak defender window in 2023: the most time n-day defenders had to patch before exploitation.

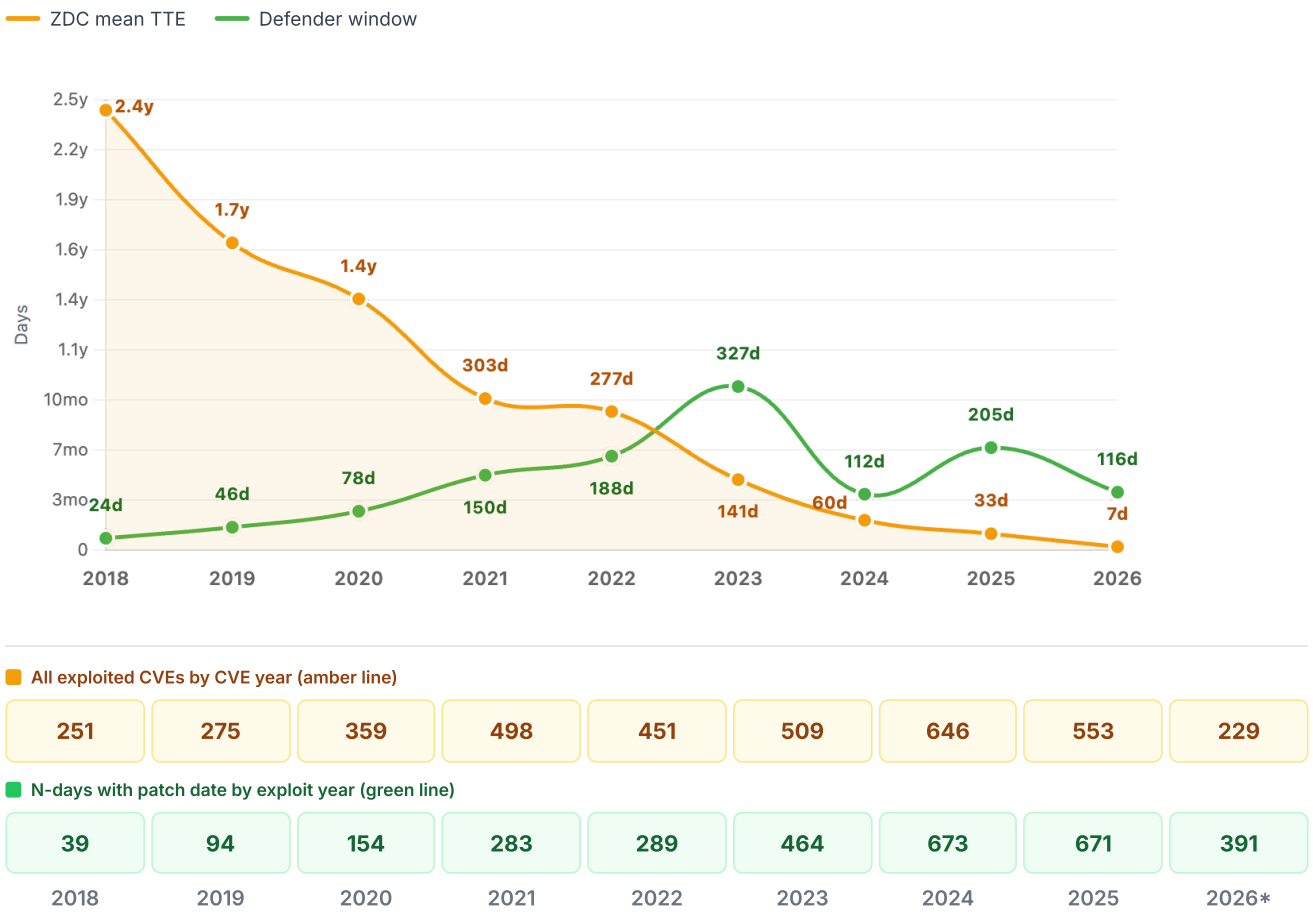
327d → 112d
2023 to 2024: the sharpest single-year defender window drop; defenders had barely a third the runway they had the year before

+11.5 d/yr
average annual increase in the defender window, January 1, 2018 through July 15, 2026: it widened from 24 days to 116, rather than collapsing

FIGURE 3

Mean Time-to-Exploit (TTE) vs. Defender Window, January 1, 2018 through July 15, 2026

Two different views of adversary speed. Our amber values closely match ZDC's published figures (2024: 60d vs ZDC's 53d). ZDC mean TTE: 10% trimmed mean, NVD pub date to first exploitation, all exploited CVEs, by CVE year. Defender window: median days from patch release to first exploitation, n-days only, by exploit year. These lines measure different gaps with different year groupings. Do not subtract one from the other. 2026 is a partial year through mid-July.



All exploited CVEs by CVE year (amber line)

2018	2019	2020	2021	2022	2023	2024	2025	2026*
251	275	359	498	451	509	646	553	229

N-days with patch date by exploit year (green line)

2018	2019	2020	2021	2022	2023	2024	2025	2026*
39	94	154	283	289	464	673	671	391

* 2026 partial year through mid-July. The two rows use different year groupings; do not subtract one from the other.

Open any security trade publication, and you'll find an apocalyptic tale: AI-armed adversaries now exploit vulnerabilities within days or even *hours* of publication. ZeroDayClock supplies that story with its central statistic, and may well have started it. If you take the amber line at face value, you would conclude defenders have days to patch, sometimes less.

SOURCE · ZERODAYCLOCK

What is TTE?

Time-to-Exploit (TTE) measures the elapsed time between when a vulnerability is publicly disclosed (NVD publication date) and when the first confirmed in-the-wild exploitation signal was observed.

```
TTE = date_exploit_signal - date_nvd_publication   (days)
Positive = after disclosure  ·  Zero = same calendar day  ·  Negative = before disclosure
```

ZeroDayClock's own definition, from zerodayclock.com/signatories.

When we start the clock at the patch instead, we see a very different picture. Adversaries have *not* sped up in eight years. They allowed defenders more than 100 days of runway every year since 2021, they are allowing 116 so far in 2026, and the trendline shows a window for defenders that has generally been tracking up at a rate of 11.5 days per year.

The two lines disagree because they measure different things. ZeroDayClock times adversaries from NVD publication, and NVD publishes late: its catalogers have lately averaged 25 days behind the vendor's patch. Every day they fall further behind makes adversaries look faster, even when adversaries haven't changed a thing they do. In 2026 the amber line touches zero largely because adversaries exploited half of the year's CVEs before NVD had published the records at all. ZeroDayClock's number blends adversary speed with NVD's backlog, and the backlog supplies most of the recent movement.

So why does the amber line say otherwise? Two effects pull it down, and neither is adversary speed. The first is NVD lag. ZeroDayClock measures from NVD publication rather than patch availability, so the longer NVD takes to catalog a CVE after its patch ships, the faster adversaries appear to move. In 2026 the amber sits near zero because half of this year's exploited CVEs were hit before NVD even published them. The two lines are timing the same events from different starting guns.

The amber line shrinks for a second reason: catalogers haven't finished counting the recent years. Imagine you are watching a marathon. Two hours in, you check the leaderboard, see eight runners across the line all under 2:15, average their times, and conclude the typical marathon takes 2 hours and 12 minutes. It is a wrong conclusion. Thousands are still on the course, uncounted, and as they finish over the next several hours the average climbs.

The recent CVE years are full of runners who haven't finished. Catalogers confirm exploitation of old vulnerabilities years after the fact, so they are still filling in the slow tails of 2024, 2025, and 2026, and as they log the stragglers they will pull those years' averages up too. Anyone who reads that half-finished record as proof of acceleration has the mechanism backwards. We went looking for AI-driven acceleration in the exploit record, and we did not find it.

Could Adversaries Actually Be Taking Longer to Exploit Vulnerabilities?

To this point we have explained the widening defender window without touching our assumptions about adversaries: defenders left more exposure standing, and catalogers introduced lag.

We based those conclusions on real evidence, but we can't ignore a third explanation: that adversaries genuinely take their time now. So before moving on, and without going so far as to claim adversaries are actually slowing down, we'll lay out four (unproven) reasons they might:

1. **Adversaries ration speed, because speed rarely pays.** When defenders leave millions of systems unpatched for years, an adversary gains almost nothing by striking on day one; the same target will still be standing in month sixty. Building fast, reliable exploits also costs real money, so adversaries spend that money on targets that justify it and let commodity scanners sweep up the neglected mass with recycled CVEs years later. Every time adversaries make that trade, they push the average first-exploitation date later.
2. **Reliable exploitation takes work.** Turning a patch into a stable exploit that works across real deployments, environment quirks, and detection evasion, is not instant. Adversaries who value reliability over speed wait until the exploit is dependable. They may also prefer to wait for someone else to do the work and buy their tool.
3. **Careful adversaries protect their assets.** A working exploit is an asset, and using it can burn the access it buys, expose the tooling, and put defenders on alert. Patient adversaries, state actors especially, hold their capability back and pick their timing and targets to stay undetected, rather than firing the moment a patch ships.
4. **Faster defenders push adversaries into the long tail.** As well-run organizations close their critical exposures quickly, what remains to exploit is increasingly the abandoned and the forgotten, which is found and hit late by definition.

Any of these four situations could be real, and the public exploit record can't yet tell us which. Consider them open questions, not conclusions.

What Zero-Days, Same-Days, and N-Days Look Like

Here, we take 2026's fastest exploitations in order: from the worst case for defenders to the best.

Worst case: adversaries struck before any patch existed, and nobody could have patched their way out. In every other case the vendor had already shipped a fix.

For n-days, SLAs can have an impact. Most enterprise patch programs treat 30 days as the deadline for critical CVEs. Adversaries hit 131 of 2026's 391 n-days (33.5%) within 30 days of the patch: some the same day the vendor published, where even teams that patch immediately fail. Here teams that enforce their SLA win. Defenders need to know which situation they're in before they pick a defense.

TRUE ZERO-DAY

Zero-Day

No patch existed when first exploited, 2026

71 CVES

COMPOSITION BY SOFTWARE TYPE

ALL CVES

WordPress 35 (49%) · Commercial 25 (35%) · OSS 11 (15%)

NON-WORDPRESS

Commercial 25 (69%) · OSS 11 (31%)

Half are WordPress plugins.
Of the non-WP zero-days, 69% are commercial enterprise products.

TOP VENDORS

17 of 71 CVEs · 24%

Microsoft	9
Fortinet	3
Google	2
Cisco	2
Apple	1

VULNERABILITY CLASSES

Memory / Other	19
Injection	15
Auth & Access	15
File & Path	13
Privilege Esc.	9

IMPLICATION

No patch existed when these were first exploited, so patch speed cannot help. Defense depends on compensating controls, attack-surface reduction, and detection that does not wait on a vendor advisory. For WordPress, the highest-leverage control is running fewer plugins.

N-DAY EXPLOITATION

Same-Day

0 DAYS

Exploit confirmed on the same calendar date as patch release, 2026 n-days

32 CVES

COMPOSITION BY SOFTWARE TYPE

ALL CVES

WordPress 11 (34%) · Commercial 14 (44%) · OSS 7 (22%)

NON-WORDPRESS

Commercial 14 (67%) · OSS 7 (33%)

67% of non-WP cases are commercial enterprise software.
Authentication bypass, deserialization, and integer overflow dominate: all categories with mature public tooling.

TOP VENDORS

10 of 32 CVEs · 31%

Microsoft	3
Ivanti	2
SonicWall	2
Cisco	2
Palo Alto	1

VULNERABILITY CLASSES

Memory / Other	8
Auth & Access	7
Injection	7
File & Path	5
Privilege Esc.	3

IMPLICATION

Standard patch SLAs provide zero protection here. The patch and the attack arrive on the same day. The only viable defense is compensating controls deployed before the advisory publishes.

N-DAY EXPLOITATION

1-30 Day

10 DAY MEAN

Actively exploited within a month of patch release, 2026 n-days

99 CVES

COMPOSITION BY SOFTWARE TYPE

ALL CVES

WordPress 13 (13%) · Commercial 37 (37%) · OSS 47 (47%)

NON-WORDPRESS

Commercial 37 (43%) · OSS 47 (55%)

55% of non-WP cases are OSS tooling.
GitHub, Apache, Langflow, NginxUI, Google. Public disclosure is the trigger, not targeted reconnaissance.

TOP VENDORS

22 of 99 CVEs · 22%

Microsoft	6
Cisco	5
SolarWinds	4
GitHub (OSS)	4
Fortinet	3

VULNERABILITY CLASSES

Injection	27
Auth & Access	18
Supply Chain	9
File & Path	9
Memory / Other	5

IMPLICATION

A 14-day SLA covers the median with room to spare. Injection and Missing Authentication both have compensating controls you can deploy while a patch is staged. The primary target is OSS tooling. You cannot patch what you do not know you are running.

Eleven of the 32 same-day CVEs are WordPress plugins. Wordfence and Patchstack find active attacks, coordinate a fix with the plugin author, and publish everything the same day, so the exploit date and the patch date always match. Those eleven tell us nothing about adversary speed.

In the other 21, adversaries really did strike the day the patch shipped: 14 commercial enterprise products (Microsoft, Cisco, Ivanti, SonicWall at the top) and 7 open-source projects. They used a scattering of flaw types: four incorrect privilege assignments, then deserialization, path traversal, and missing authentication twice each. No SLA helps in this tier. The only way to defend against these was to have controls in place that prevented damage or stopped the bleeding.

The 1-30 day tier looks different. Adversaries went after open source 47% of the time, up from 22% among the same-days, plus another 13% WordPress, putting 60% of the tier outside commercial software. Missing authentication was their most-used flaw: ten CVEs, all outside WordPress. An auth bypass takes almost nothing to weaponize; the adversary walks through a door that never checks credentials. In the meantime, before the patch rolls out, a WAF rule or a network ACL can block most injection and missing-auth attacks outright.

So why two tiers? Because they belong to two different kinds of attacker.

Actors with pre-patch access hit commercial infrastructure on day zero or earlier. Opportunists watch disclosures and scan open-source software over the following weeks. Nobody patches their way past the first group: those defenses have to exist beforehand. Teams can (and do) patch their way past the second, if they actually hit their SLA dates.

Different situations require different defenses. There's no one policy that catches them all.

True Zero-Days: What the Data Actually Shows

Defenders fear zero-days for a real reason: until the vendor ships a fix, they can't patch their way out. But the industry also misunderstands zero-days, because reporters and vendors apply the term loosely to nearly any high-profile CVE in the news, which inflates how common they seem and buries what adversaries actually did. That loose usage may even explain how the Zero Day Clock ended up with the wrong definition, counting any CVE exploited on or before its NVD publication date as a zero-day, whether or not a patch already existed.

An adversary who wants a genuine zero-day has to find a flaw nobody else has found, ideally in widely deployed software, and build a working exploit before the vendor knows the flaw exists. Most adversaries never bother, because n-days get them the same access at a fraction of the effort: the vulnerability is public, the patch often comes with hints, someone may have published exploit code, and enough targets stay unpatched long enough to make the whole exercise reliable.

Many people understandably believe AI puts that calculus into question. If AI tools cut the skill and time an adversary needs to find flaws and build exploits, zero-days stop being scarce. So we put eight years of confirmed exploitation to the questions:

- How big is the zero-day problem?
- Where do adversaries concentrate it?
- Has AI measurably moved it?

The record answers more definitively than the Vulnpocalypse headlines suggest.

Throughout, we say "true zero-day" to keep our metric separate from ZeroDayClock's. ZDC counts any exploited CVE recorded on or before NVD's publication date. We apply the definition practitioners have used for decades: adversaries struck before a patch existed.

SOURCE · ZERODAYCLOCK

Zero-Day Classification

Zero-day: TTE ≤ 0 — exploitation confirmed on the same calendar day as disclosure or earlier. Includes genuine pre-disclosure (TTE < 0) and same-day events (TTE = 0) where intra-day order cannot be determined at date-only timestamp precision. Overall zero-day rate: **37%** all years; **68%** in 2026.

ZeroDayClock's own definition, from zerodayclock.com/signatories.

711	0.28%	1 in 5	2025	~10/mo
total true zero-days out of 3,769 classified CVEs with confirmed exploitation.	(711) of all CVEs published since 2018 are true zero-days	of all known-exploited CVEs (3,769) are true zero-days	was the peak year on record with 143 true zero-days first exploited	true zero-days on average during 2026 (71 YTD through mid-July)

ZeroDayClock's headline number rises and falls with NVD's workload, not with anything adversaries do. Suppose NVD's catalogers slipped from 25 days behind the vendor's patch to 35. ZeroDayClock would relabel another 364 CVEs as "zero-days," pushing its rate from 33.0% to 42.9%, while every adversary on earth did exactly what they did before. Our rate would sit still at 18.9%, because we don't rely on NVD's calendar.

FIGURE 4

Hypothetical Scenario (not real data): the effect of NVD lag on ZDC's rate, by year.

ZDC rate = time-to-exploit (TTE) ≤ 0 (exploit_date ≤ nvd_pub_date). Change = additional reclassified as "zero-days" if NVD lag increased by 10 days. 2026 covers only 6 months of data, so its figures are preliminary and should not be compared to full-year rows.

Year	CVEs	ZDC rate (25d NVD lag)	ZDC rate (35d NVD lag)	Change
2024	646	43.7%	55.4%	+11.8 percentage points
2025	553	43.9%	57.5%	+13.6 percentage points
2026 YTD	229	49.8%	71.6%	+21.8 percentage points

In that scenario, NVD's slowdown would hand ZDC about 13 points of its 2025 rate. Adversaries wouldn't have changed anything. They'd just be exploiting CVEs that NVD hadn't gotten to yet.

Why does the difference matter? Because defenders reading these numbers need to know which problem they have.

When a team gets breached through a CVE that had a patch, the team failed to apply it. An operational failure has an operational fix. When adversaries strike before any patch exists, the best-run patching program in the world would have made no difference. ZDC's metric pools those two situations under one label, so a defender reading it can't tell how much of the number they could have prevented.

We ran ZDC's method over our own 3,769 CVEs and got 33.0%. We then applied the practitioner definition to the same CVEs and got 18.9%. Nothing separates those two figures (33.0% and 18.9%) except NVD's publishing delays: 14.1 points of "zero-days" that were actually patched CVEs waiting in NVD's queue.

TL;DR: ZeroDayClock measures how quickly NVD publishes records, and when NVD slows down, ZeroDayClock reports that adversaries sped up. Anyone who wants to understand real-world exploitation should set it aside.

The "True" Zero-Day Population

Any investigation into zero-days has to start with scale. Figure 5 maps the full population of 253,912 CVEs published since 2018 against four outcomes. A CVE is either never exploited, exploited as an n-day where a patch existed, a true zero-day in non-WordPress software, or a true zero-day in a WordPress plugin. That context matters before examining

where zero-days concentrate, which vendors produce them, and how the count has shifted over time. WordPress plugin CVEs are highlighted because they are large enough to influence the overall dataset.

FIGURE 5

253,912 CVEs Published (January 1, 2018 through July 15, 2026): 1.48% Were Exploited, 0.28% Were True Zero-Days

Gray = never exploited (98.5%). Amber = n-days: patch existed before exploitation (1.2%). Red + purple = true zero-days: no patch existed at time of exploitation (0.28%).

True zero-days, non-WordPress: 503
No patch existed at time of exploitation; commercial software, enterprise, browsers, mobile

True zero-days, WordPress plugins: 208
No patch existed at time of exploitation; WordPress plugin CVEs (Wordfence, Patchstack, WPScan)

Other KEVs (patch existed): 3,058
N-days: a patch was available before exploitation was first confirmed

Never exploited: 250,143
CVEs published since 2018 with no confirmed in-the-wild exploitation on record (98.5%)

Each square = 200 CVEs. Red + amber = all 3,769 known-exploited CVEs (KEV, 1.48% of all CVEs). Total population: 253,912 CVEs.

In Figure 5 we map all 253,912 CVEs published since 2018 into four outcomes, in which adversaries:

- Never touched the CVE
- Exploited the CVE after a patch existed
- Exploited the CVE pre-patch in ordinary software
- Exploited the CVE pre-patch in a WordPress plugin

We separate out the WordPress plugins because they are numerous enough to move the totals on their own and differ in kind from the rest, mass-market plugin flaws rather than the enterprise and platform bugs adversaries prize most. Breaking them out lets us see whether the zero-day picture holds once they are set aside.

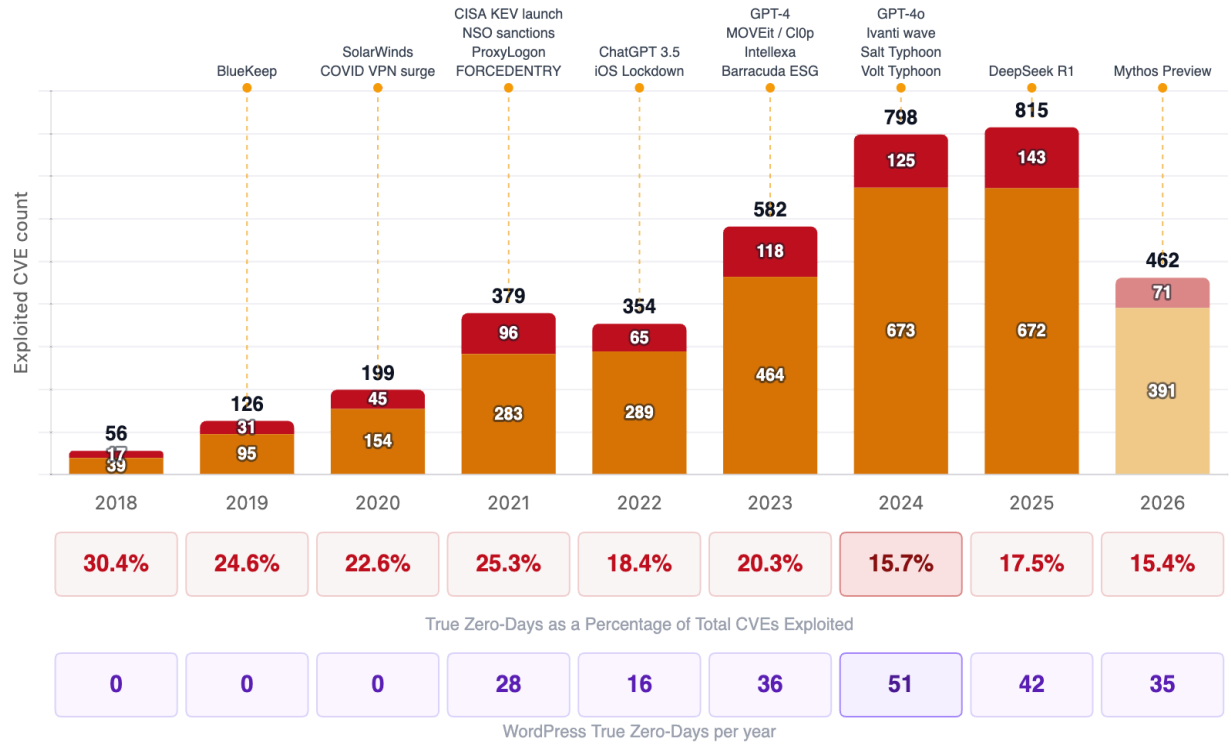
Start with the proportions. Adversaries have ignored 250,143 of those 253,912 CVEs — so far, anyway — and even the 3,769 they did exploit occupy only a narrow band of the chart. The 711 they struck before a patch existed come to about three squares in every thousand. You have to squint.

Every scanner, risk dashboard, and quarterly security deck treats CVE volume as if it measured danger, which is, frankly, *wrong*. Adversaries bear this out: year after year, they weaponize a small fraction of what researchers publish, and they strike pre-patch on a far smaller fraction still.

FIGURE 6

Exploited CVEs by Year of First Exploitation (January 1, 2018 through July 15, 2026): True Zero-Days vs. N-Days

Bar height = total CVEs first exploited that year, stacked into true zero-days (red) and n-days (amber); n-days = exploited minus true zero-days. Stat cards below: the true zero-day rate as a share of that year's exploited CVEs (red), and the WordPress-plugin share within that year's true zero-days (purple). Yearly totals include two confirmed-exploited CVEs with no patch date (one in 2019, one in 2025); they sit in the amber n-day segments but are not part of the 3,058 n-day figure reported elsewhere.



Adversaries changed in 2021 and never changed back. They went from 45 true zero-days in 2020 to 96 in 2021, and (apart from a dip in 2022) they stayed above that level through 2025. You might suspect a cataloging artifact here, since CISA launched its KEV catalog in November 2021, but CISA assigned its retroactive entries to their original years, so that can't explain the jump. 2021 really did represent a massive change: adversaries nearly tripled their attacks on Apple, from 5 zero-days to 14, Microsoft began admitting active attacks on patch day, and WordPress plugins went from zero recorded zero-days to 28.

Adversaries set the zero-day record in 2025, exploiting 143 true zero-days, ahead of 125 in 2024 and 118 in 2023. Take the WordPress plugins out of the count and 2025 still wins: 101 vs. 82 and 74. While earlier peak years were mostly WordPress volume, in 2025, adversaries did peak numbers against real enterprise and mobile targets.

So far, 2026 has stayed below that peak, with 71 true zero-days through mid-July (36 outside WordPress), on pace for roughly 131 and 66 for the full year, both under 2025's marks.

Everyone predicted AI-assisted vulnerability research would hand adversaries a surge. Halfway through the year, the adversaries haven't shown that to be true.

Raw counts only tell half the story, though, because researchers publish more CVEs every year. So we also measured the rate: out of everything adversaries exploited in a given year, what percentage did they strike before a patch existed?

If AI were pushing adversaries toward zero-days, we'd expect that percentage to climb. It hasn't.

In every year since 2021, adversaries struck pre-patch on somewhere between 15.4% and 25.3% of what they exploited, roughly one CVE in five, and 2021 itself was the highest of those years.

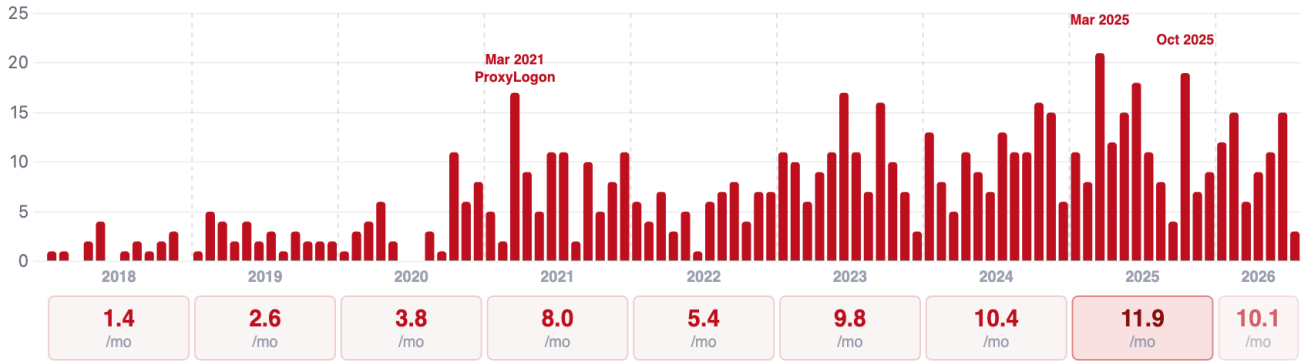
Go back further, and the percentages ran higher still. In 2018, when adversaries spent their effort on browsers and phones, 30.4% of what they exploited had no patch yet, then 24.6% in 2019 and 22.6% in 2020, and 25.3% in 2021. So when the zero-day count doubled in 2021, the percentage barely moved, and after 2021 it mostly fell: 18.4%, then 20.3%, then 15.7%, then 17.5%, because adversaries were exploiting so many more n-days and WordPress plugins that the zero-day portion shrank in comparison.

Measured this way, adversaries relied on zero-days *less* in the last two years than in any year we studied. Whatever AI has done for the bad guys, it has not made them reach for zero-days more often.

FIGURE 7

True Zero-Days by Month (Jan 2018–Jul 2026): March 2025 (21) and October 2025 (19) Are the Two Highest Months on Record

The annual bars in Figure 6 smooth over within-year variation. The monthly view reveals the spikes. March 2025 (21 TZDs), October 2025 (19), and June 2025 (18) stand out as the three highest individual months on record, with March 2021 (ProxyLogon, 17) and June 2023 (17) close behind. Each bar is one calendar month. Year labels mark January of each year. 711 total TZDs across 103 months; months with zero TZDs are shown as gaps.



Each bar is one calendar month. Year labels mark January of each year. 711 total TZDs across 103 months; months with zero TZDs are shown as gaps.

The annual bars in Figure 6 also smooth over how adversaries actually work within a year, so Figure 7 breaks the same record into months, and the months show the 2021 shift even more clearly. Before 2021, adversaries produced zero to three true zero-days in a typical month, and their biggest month on record was October 2020, with 11. Then in March 2021 they exploited 17 in a single month, most of that ProxyLogon, and from then on double-digit months became the "new normal."

The monthly view also shows that adversaries work in bursts. Across the 103 months we studied, they left some months completely empty, but packed others with 20 or more. Why? Because when adversaries find a productive attack surface, they exploit its variations quickly, then move on once the opportunity closes. March 2025, with 21, and

October 2025, with 19, are the two biggest bursts on record, and June 2025 sits just behind them at 18.

So we've seen the trends on zero-days, in both absolute and relative terms. Next, we ask what these zero-days actually are: whose software adversaries went after, which kinds of flaws they used, and how the commercial and open-source worlds compare.

Which Software Vendors Have the Most True Zero-Days

Figure 8 sorts the true zero-days by vendor. We attribute each CVE to its CVE Numbering Authority, the organization authorized to assign and publish the CVE record, which in most cases is the vendor whose software had the flaw. Where no CNA record exists, we identify the vendor from the affected product.

Why bother sorting it this way? Because when a short list of vendors accounts for most pre-patch exploitation, defenders can focus more resources on that same short list, instead of spreading their scrutiny evenly.

29%

of all true zero-days are in WordPress plugins (208), the single largest software category

57

CVE Numbering Authorities in the dataset have had at least one true zero-day

66%

of all true zero-days come from WordPress plugins and five named vendors: Microsoft, Apple, Google, Cisco, and Fortinet

WordPress

Plugins exceeded any individual named vendor in true zero-day count every year from 2021 onward

FIGURE 8

True Zero-Days by CVE Numbering Authority (January 1, 2018 through July 15, 2026): WordPress Plugin CNAs Are 29% of All Cases; Microsoft, Apple, and Google Follow

Each cell shows true zero-day count (top) and share of all true zero-days that year (bottom). Linear scale. Faded = still counting (2025-2026). Totals: WordPress Plugins 208, Microsoft 132, Apple 64, Google 38, Fortinet 15, Cisco 15, Samsung 11, Mozilla 6, Apache 4, VMware 4, Adobe 3, Qualcomm 3.

Lower % — Higher % | Linear scale. Faded = still counting (2025–2026)

	2018	2019	2020	2021	2022	2023	2024	2025	2026	Total
WordPress Plugins				29.2% 28	24.6% 16	30.5% 36	40.8% 51	29.4% 42	49.3% 35	208
Microsoft	41.2% 7	35.5% 11	17.8% 8	21.9% 21	16.9% 11	16.9% 20	18.4% 23	15.4% 22	12.7% 9	132
Apple			11.1% 5	14.6% 14	18.5% 12	17.8% 21	4.8% 6	3.5% 5	1.4% 1	64
Google		9.7% 3	15.6% 7	12.5% 12	6.2% 4	4.2% 5	2.4% 3	1.4% 2	2.8% 2	38
Fortinet				1.0% 1	3.1% 2	0.8% 1	2.4% 3	3.5% 5	4.2% 3	15
Cisco			2.2% 1		1.5% 1	3.4% 4	2.4% 3	2.8% 4	2.8% 2	15
Samsung			15.6% 7	2.1% 2	1.5% 1			0.7% 1		11
Mozilla		6.5% 2	6.7% 3				0.8% 1			6
Apache				3.1% 3	1.5% 1					4
VMware							0.8% 1	2.1% 3		4
Adobe	11.8% 2								1.4% 1	3
Qualcomm			4.4% 2				0.8% 1			3
Total	17	31	45	96	65	118	125	143	71	

Apple deserves the closest look. Adversaries exploited 5 of its zero-days in 2020, then 14 in 2021, then 21 in 2023 at the peak. After that they mostly quit: 6 in 2024, 5 in 2025. Google got a smaller version of the same treatment, 12 in 2021 and then 2 to 5 a year through 2025, nearly all of it Chrome.

WordPress plugins are the biggest single category, 208 of the 711, or 29%, and they don't belong in the same conversation as the rest. Wordfence and Patchstack find these flaws while adversaries are already using them, then publish the disclosure and the patch together on the same day.

Microsoft is at or near the top of the list consistently, roughly 20 to 23 true zero-days per year in 2021, 2023, 2024, and 2025, with 2022 as the exception at 11. That reflects continued targeting of Microsoft products and Microsoft's practice of acknowledging known exploitation on patch day, ensuring their cases appear in the data rather than being quietly fixed.

Samsung shows a concentrated burst, with 7 zero-days first exploited in 2020 and 2 in 2021, then near-zero every year after. That pattern appears at smaller scale across other vendors throughout the dataset. Adversaries find a productive target, exploit the available attack variations, and move on.

Which Vulnerability Classes Drive the Most True Zero-Days

A CWE (Common Weakness Enumeration) identifies the type of flaw that let adversaries in. Use-after-free means the software kept using memory it had already given back. OS command injection means an adversary tricked the software into running commands of the attacker's choosing. Figure 9 maps every true zero-day to its CWE, and here adversaries concentrate just as hard as they did by vendor: fifteen weakness types cover 59% of everything they did, out of 944 potential weaknesses that exist.

OOB Write
is the #1 vulnerability class among true zero-days overall (62 cases)

XSS
is the #1 vulnerability class within WordPress true zero-days

15 CWEs
cover 59% of true zero-day vulnerability classes

103 of 944
weakness types (CWEs) have ever produced a true zero-day

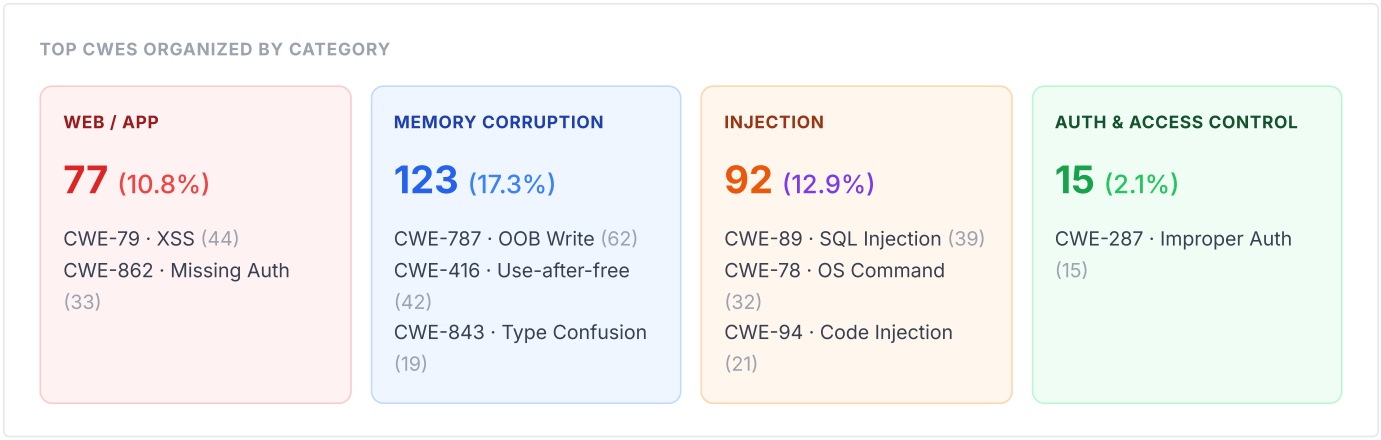
FIGURE 9

True Zero-Days by Vulnerability Class (January 1, 2018 through July 15, 2026): OOB Write Leads Overall; XSS Leads Within WordPress Plugin CVEs

Linear scale. Faded = still counting (2025-2026). Leading totals: CWE-787 Out-of-bounds Write 62, CWE-79 Cross-site Scripting 44, CWE-416 Use After Free 42, CWE-89 SQL Injection 39, CWE-434 Unrestricted File Upload 35, CWE-862 Missing Authorization 33, CWE-78 OS Command Injection 32, CWE-22 Path Traversal 24, CWE-94 Code Injection 21, CWE-843 Type Confusion 19.

Lower % — Higher % | Linear scale. Faded = still counting (2025–2026)

	2018	2019	2020	2021	2022	2023	2024	2025	2026	Total
CWE-787 Out-of-bounds Write	5.9% 1		33.3% 15	10.4% 10	20.0% 13	8.5% 10	5.6% 7	3.5% 5	1.4% 1	62
CWE-79 Cross-site Scripting	5.9% 1	6.5% 2	6.7% 3	8.3% 8	9.2% 6	3.4% 4	9.6% 12	2.1% 3	7.0% 5	44
CWE-416 Use After Free	11.8% 2	12.9% 4	8.9% 4	9.4% 9	1.5% 1	4.2% 5	5.6% 7	6.3% 9	1.4% 1	42
CWE-89 SQL Injection		3.2% 1	4.4% 2	11.5% 11	1.5% 1	2.5% 3	8.8% 11	4.9% 7	4.2% 3	39
CWE-434 Unrestricted File Upload			2.2% 1	4.2% 4	3.1% 2	1.7% 2	5.6% 7	6.3% 9	14.1% 10	35
CWE-862 Missing Authorization		3.2% 1		6.3% 6	6.2% 4	2.5% 3	7.2% 9	4.9% 7	4.2% 3	33
CWE-78 OS Command Injection		9.7% 3		1.0% 1	1.5% 1	2.5% 3	4.8% 6	11.2% 16	2.8% 2	32
CWE-22 Path Traversal		9.7% 3		5.2% 5	3.1% 2	2.5% 3	1.6% 2	5.6% 8	1.4% 1	24
CWE-94 Code Injection		3.2% 1			3.1% 2	5.1% 6	4.0% 5	2.8% 4	4.2% 3	21
CWE-843 Type Confusion			6.7% 3	2.1% 2	4.6% 3	4.2% 5	2.4% 3	1.4% 2	1.4% 1	19
CWE-287 Improper Authentication				2.1% 2	3.1% 2	1.7% 2	4.0% 5	0.7% 1	4.2% 3	15
CWE-20 Improper Input Validation	11.8% 2			1.0% 1	3.1% 2	4.2% 5	1.6% 2	1.4% 2	1.4% 1	15
CWE-269 Improper Privilege Management			4.4% 2		1.5% 1		1.6% 2	2.1% 3	8.5% 6	14
CWE-306 Missing Authentication			2.2% 1	2.1% 2	3.1% 2	0.8% 1	2.4% 3	1.4% 2	4.2% 3	14
CWE-502 Deserialization of Untrusted Data				1.0% 1	3.1% 2		1.6% 2	3.5% 5		10
Total	17	31	45	96	65	118	125	143	71	



Out-of-bounds write leads the list at 62 cases. XSS and missing authorization rank just behind it, due to WordPress plugins. Plugin authors keep accepting user input without checking it and shipping access controls that don't enforce what they claim, so adversaries keep exploiting both, year after year.

The memory-corruption flaws, out-of-bounds write, use-after-free, and type confusion, belong to the browser-and-mobile era. Adversaries used them most heavily from 2020 through 2023, exactly the years they hit Apple and Chrome hardest, and they rarely used one alone. A typical campaign chained several: one flaw to compromise the browser, another to escape its security layer, and a third to take over the device.

Injection and authentication bypass belong to the enterprise era. From 2022 onward, adversaries drove the rise in enterprise exploitation with OS command injection, SQL injection, code injection, and improper authentication.

Vulnerability Classes: WordPress-Only True Zero-Days

Within WordPress, adversaries spread their work almost evenly across four kinds of flaws: XSS (29), unrestricted file upload (28), missing authorization (27), and SQL injection (24), which together account for more than half of all WordPress true zero-days. All four grow from the same bad habits. Plugin authors accept user input without checking it, handle files without enforcing types, and write access controls that don't verify what they promise. When adversaries surged against WordPress in 2024, XSS and SQL injection climbed right along with the volume, 9 and 10 cases apiece.

FIGURE 10

WordPress Plugin True Zero-Days by Vulnerability Class (January 1, 2018 through July 15, 2026): XSS and Missing Authorization Account for the Majority

Lower % — Higher % | Linear scale. Faded = still counting (2025–2026)

	2018	2019	2020	2021	2022	2023	2024	2025	2026	Total
CWE-79 Cross-site Scripting				21.4% 6	37.5% 6	8.3% 3	17.6% 9	2.4% 1	11.4% 4	29
CWE-434 Unrestricted File Upload				14.3% 4	12.5% 2	5.6% 2	13.7% 7	14.3% 6	20.0% 7	28
CWE-862 Missing Authorization				10.7% 3	25.0% 4	8.3% 3	15.7% 8	14.3% 6	8.6% 3	27
CWE-89 SQL Injection				28.6% 8			19.6% 10	7.1% 3	8.6% 3	24
CWE-94 Code Injection						11.1% 4	3.9% 2	7.1% 3	2.9% 1	10
CWE-288 Auth Bypass via Alternate Path						13.9% 5	2.0% 1	9.5% 4		10
CWE-269 Improper Privilege Management					6.3% 1		2.0% 1	7.1% 3	11.4% 4	9
CWE-287 Improper Authentication				7.1% 2	6.3% 1	2.8% 1	2.0% 1		5.7% 2	7
CWE-22 Path Traversal				3.6% 1				9.5% 4	2.9% 1	6
CWE-98 PHP Remote File Inclusion							3.9% 2	7.1% 3	2.9% 1	6
CWE-284 Improper Access Control						13.9% 5				5
CWE-306 Missing Authentication				3.6% 1			3.9% 2		2.9% 1	4
CWE-502 Deserialization of Untrusted Data					6.3% 1		2.0% 1	4.8% 2		4
CWE-639 IDOR / Insecure Direct Object Ref.						2.8% 1	2.0% 1		5.7% 2	4
No CWE Assigned						2.8% 1	2.0% 1	2.4% 1		3
Total	**0**	**0**	**0**	**28**	**16**	**36**	**51**	**42**	**35**	

WordPress plugin true zero-days only. Share of that year's WordPress plugin TZDs, with the CVE count as the subscript. 2026 is a partial year.

Vulnerability Classes: Non-WordPress True Zero-Days

Take WordPress out and the list looks different. XSS and missing authorization drop out of the top positions, and what's left is the work adversaries spend real money on: memory corruption and injection flaws in commercial software, enterprise products, browsers, and mobile platforms.

FIGURE 11

Non-WordPress True Zero-Days by Vulnerability Class (January 1, 2018 through July 15, 2026): Memory Corruption and Injection Are the Dominant Threat Classes

Lower % — Higher % | Linear scale. Faded = still counting (2025–2026)

	2018	2019	2020	2021	2022	2023	2024	2025	2026	Total
CWE-787 Out-of-bounds Write	5.9% 1		33.3% 15	14.7% 10	26.5% 13	12.2% 10	9.5% 7	5.0% 5	2.8% 1	62
No CWE Assigned	23.5% 4	38.7% 12	4.4% 2	17.6% 12	12.2% 6	13.4% 11	1.4% 1	2.0% 2		50
CWE-416 Use After Free	11.8% 2	12.9% 4	8.9% 4	13.2% 9	2.0% 1	6.1% 5	9.5% 7	8.9% 9	2.8% 1	42
CWE-78 OS Command Injection		9.7% 3		1.5% 1	2.0% 1	3.7% 3	8.1% 6	15.8% 16	5.6% 2	32
CWE-843 Type Confusion			6.7% 3	2.9% 2	6.1% 3	6.1% 5	4.1% 3	2.0% 2	2.8% 1	19
CWE-22 Path Traversal		9.7% 3		5.9% 4	4.1% 2	3.7% 3	2.7% 2	4.0% 4		18
CWE-20 Improper Input Validation	11.8% 2			1.5% 1	4.1% 2	6.1% 5	2.7% 2	2.0% 2	2.8% 1	15
CWE-79 Cross-site Scripting	5.9% 1	6.5% 2	6.7% 3	2.9% 2		1.2% 1	4.1% 3	2.0% 2	2.8% 1	15
CWE-89 SQL Injection		3.2% 1	4.4% 2	4.4% 3	2.0% 1	3.7% 3	1.4% 1	4.0% 4		15
CWE-94 Code Injection		3.2% 1			4.1% 2	2.4% 2	4.1% 3	1.0% 1	5.6% 2	11
CWE-306 Missing Authentication			2.2% 1	1.5% 1	4.1% 2	1.2% 1	1.4% 1	2.0% 2	5.6% 2	10
CWE-125 Out-of-bounds Read				1.5% 1	2.0% 1	3.7% 3	1.4% 1	3.0% 3	2.8% 1	10
CWE-119 Buffer Overflow	5.9% 1			2.9% 2	2.0% 1	1.2% 1	1.4% 1	2.0% 2	2.8% 1	9
CWE-863					4.1% 2	4.9% 4		2.0% 2		8
CWE-287 Improper Authentication					2.0% 1	1.2% 1	5.4% 4	1.0% 1	2.8% 1	8
Total	17	31	45	68	49	82	74	101	36	

Non-WordPress true zero-days only. Share of that year's non-WordPress TZDs, with the CVE count as the subscript. 2026 is a partial year.

Open Source vs. Commercial Software

Commercial software and open source reach defenders differently: teams hear about the flaw through different channels, receive the patch on different timelines, and get different amounts of warning before adversaries arrive. So Figures 12 and 13 split the true zero-days three ways: commercial, WordPress plugins, and everything else in open source. We separate the three because WordPress wrecks any combined open-source number. Count it, and open source beat commercial in 2021, 2024, and 2026 so far. Set it aside, and adversaries barely touched the rest of open source, holding it below commercial in every single year.

3 of 9

years since 2018 where open source software had more true zero-days than commercial software

2022

Commercial regained the lead as WP plateaued. Within OSS, WordPress was already the majority category. Non-WP at its smallest relative share. The direction was set.

2024

OSS led overall for the second time (after 2021), driven almost entirely by WordPress: 51 of the 69 OSS zero-days that year were WordPress plugins, with non-WP OSS contributing just 18.

58%

(208 of 359) of all true zero-days in open source software are WordPress plugins (January 1, 2018 through July 15, 2026)

FIGURE 12

Commercial vs. Open Source True Zero-Days (January 1, 2018 through July 15, 2026): Commercial Leads Most Years; OSS Leads in 2021 (53%), 2024 (55%), and 2026 YTD (65%)

True zero-day counts by year of first exploitation. The dashed segment marks 2026 (year-to-date, partial).

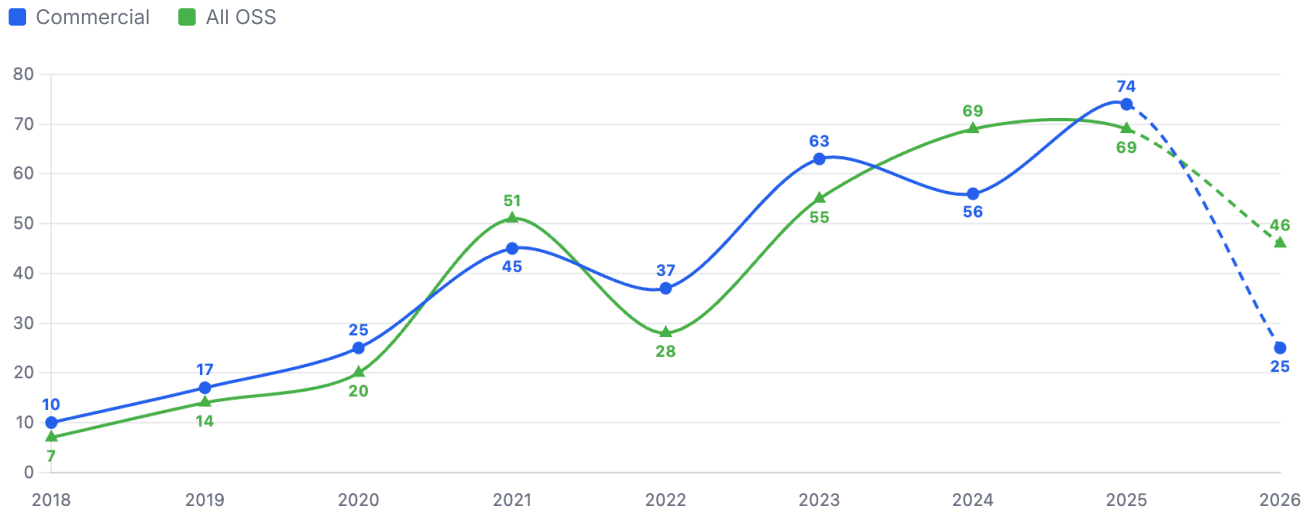
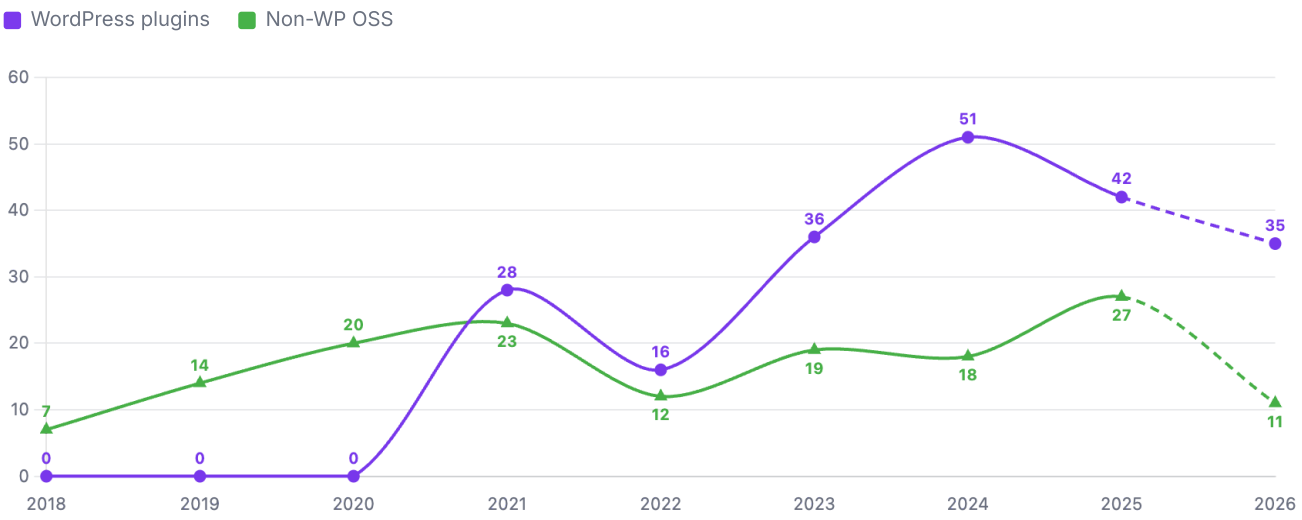


FIGURE 13

True Zero-Days Within Open Source: WordPress Plugins vs. Non-WordPress OSS (January 1, 2018 through July 15, 2026)

True zero-day counts by year of first exploitation. The dashed segment marks 2026 (year-to-date, partial).



Most of the open-source total belongs to WordPress. Across the whole period, adversaries exploited between 7 and 27 non-WordPress open-source zero-days a year. Against WordPress plugins, the catalogs show zero pre-patch exploitation through 2020, then 16 to 51 cases a year from 2021 onward, once Wordfence and Patchstack started catching attacks at scale. Remember what those numbers measure, though: Wordfence and Patchstack disclose and patch in the same moment, so defenders never get a window on these CVEs.

Commercial software followed its own line. Adversaries exploited 10 commercial zero-days in 2018, 63 in 2023, and 74 in 2025, more than half the annual total in most years. Commercial and WordPress both climbed after 2021, for unrelated reasons, while the rest of open source never climbed at all. We keep the three lines separate so you can see which one actually grew.

The Bottom Line for Zero-Days

How large and concentrated is the true zero-day problem?

Small, and adversaries keep it that way. Across eight years they returned to the same few high-value targets: browsers, mobile platforms, enterprise perimeter products, and, in a category of its own, WordPress plugins. Take WordPress out and the list shrinks to a handful of vendors whose software runs in enough places to be worth what an adversary gives up by burning a zero-day.

Has AI advancement measurably changed it so far?

If AI has changed how adversaries produce zero-days, we can't find it in the record. Adversaries kept their zero-day rate steady through exactly the years when everyone talked about AI-assisted exploitation the most. They did change one thing in that period, moving from browsers and mobile toward enterprise perimeter products, but they started that move before AI tooling was a documented factor, and a simpler explanation

works just as well: Apple and Google hardened their platforms, so adversaries went looking for softer targets. Otherwise adversaries kept hitting the same vendors, with the same kinds of flaws, on the same timelines they always have.

What do the vulnerability data trends suggest?

Adversaries have been trading memory corruption for authentication bypass and code injection, flaws that take less skill to use, so more adversaries can use them. They also traded targets, leaving browsers and mobile for enterprise perimeter products, which vendors deploy widely, update slowly, and defend with a fraction of the staff Apple or Google can afford. Both changes were underway before AI tooling counted for much.

Calibrating AI risk: possibility is not probability

The Vulnpocalypse forecasts all rest on one assumption: when exploitation gets cheaper, adversaries will do more of it. The pundits treat capability as destiny, and the scanner vendors are happy to agree, since every new CVE makes their dashboards look more essential.

AI tools really can lower the cost of finding flaws and building exploits; we don't dispute that. What we can add is a decade of evidence about what adversaries do once exploitation is within reach, and the answer is that they stay picky. They concentrated on a short list of vendors and flaw classes and on the software with the widest deployment, and they kept those habits while researchers multiplied the CVE count and the technical barriers dropped.

So far, AI is growing the catalog of what somebody could attack by tens of thousands of CVEs a year, and adversaries keep declining nearly all of it. Researchers using AI are piling more hay on the stack without adding a single needle. The dangerous CVEs were already in there, and adversaries keep finding them; the extra hay mostly buries them for everyone else. Plan your defenses around what adversaries do. Cheaper tooling has not broken their habit of sticking with what works until something forces them off it.

N-Days: The Defender Window

Adversaries exploited 711 of this dataset's 3,769 classified CVEs before any patch existed, and we covered those in True Zero-Days. Now we turn to the other 3,058: the n-days, where the vendor shipped a fix and adversaries got in anyway. (Catalogers confirmed two more exploited CVEs that carry no patch date at all, so we set those aside; see the Data Methodology.)

One question dominates this whole section: how much time did defenders have to use a fix?

Security teams have started worrying that AI is cutting that time down, so we used the eight years of confirmed exploitation, 2018 through mid-2026, to answer three questions that can help us understand the real trend:

- How much time defenders actually got between the patch and the first attack
- Whether that time differs predictably by software category and flaw type
- Whether adversaries are getting faster in any way the catalogs can detect

The answers surprised us.

CVE Volume Is Rising. Exploitation Scope Is Not.

NVD logged about 17,800 CVEs in 2018 and over 44,800 in 2025, roughly 2.5x growth in seven years. Every scanner, patch cycle, and risk dashboard passes that growth straight through to the security team as more findings, more urgency, and more noise. Adversaries apparently didn't get the memo: they're still exploiting the same thin slice of the pie they always have.

253,912
CVEs published January 1, 2018 through July 15, 2026; 31,174 through mid-July 2026, up from 17,817 in 2018

~98.5%
of all CVEs published each year have yet to be seen exploited in the wild

~1.5%
overall CVE exploitation rate; each year's newly-exploited share stayed below 2.2% even as CVE volume grew 2.5x

3,058
n-day CVEs with confirmed in-the-wild exploitation, January 1, 2018 through July 15, 2026: patch existed before exploitation was detected

~53/mo
median new exploited CVEs confirmed per month across CISA KEV and VulnCheck KEV (2022 to Jun 2026)

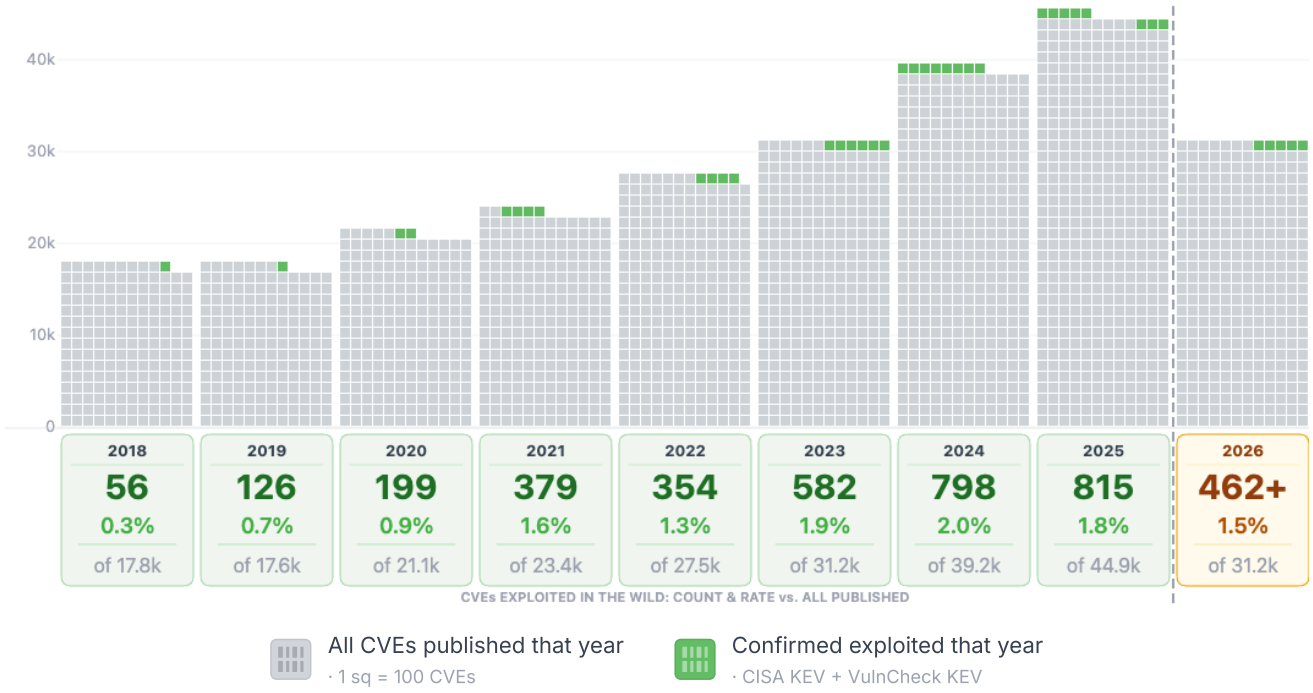
Figure 14 asks whether adversaries scale their targeting with CVE volume. Each column represents one calendar year, 2018 through mid-July 2026. Gray squares count the CVEs NVD published that year, while green squares count the ones adversaries verifiably exploited, per CISA KEV and VulnCheck KEV.

FIGURE 14

CVEs Published vs. First Confirmed Exploitation in the Wild, by Year, January 1, 2018 through July 15, 2026

The green label above each column shows confirmed exploitations and their share of all CVEs published, by year. 2026 is a partial year through mid-July.

3,769
CONFIRMED EXPLOITED
out of
253,912
CVES PUBLISHED JANUARY 1, 2018 THROUGH JULY 15, 2026



The catalog grew from 17,800 CVEs a year to nearly 45,000, and adversaries exploited between 1.3% and 2.0% of each year's CVEs, every year from 2021 through 2025. (The earlier years run lower, at least in part, because the catalogs were still building out their coverage.) So far in 2026, catalogers have confirmed 462 exploited CVEs through mid-July, and they will keep raising that number for years as they log the stragglers.

If AI tools were speeding up exploitation at scale, this chart would show two things: CVE volume breaking above its trend around 2023, and a climbing exploitation rate. It shows neither.

Researchers keep publishing along the same growth path they were on before LLMs existed. One caveat: the most capable AI tools only became widely available in 2024 and 2025, and catalogers confirm exploitation late. So we'll keep watching.

Four in five exploited CVEs already had a patch

We already know from our section on True Zero-Days that adversaries strike pre-patch about one time in five. Turn that around, and you get an n-day ratio: four out of five times, adversaries used a CVE that defenders could have patched. Figure 15 shows that ratio holding year after year, which means the n-day problem, the one defenders can actually do something about, has stayed the whole game.

81.1%
n-days: a patch was already available at time of first exploitation

4.3×
more n-days than true zero-days.

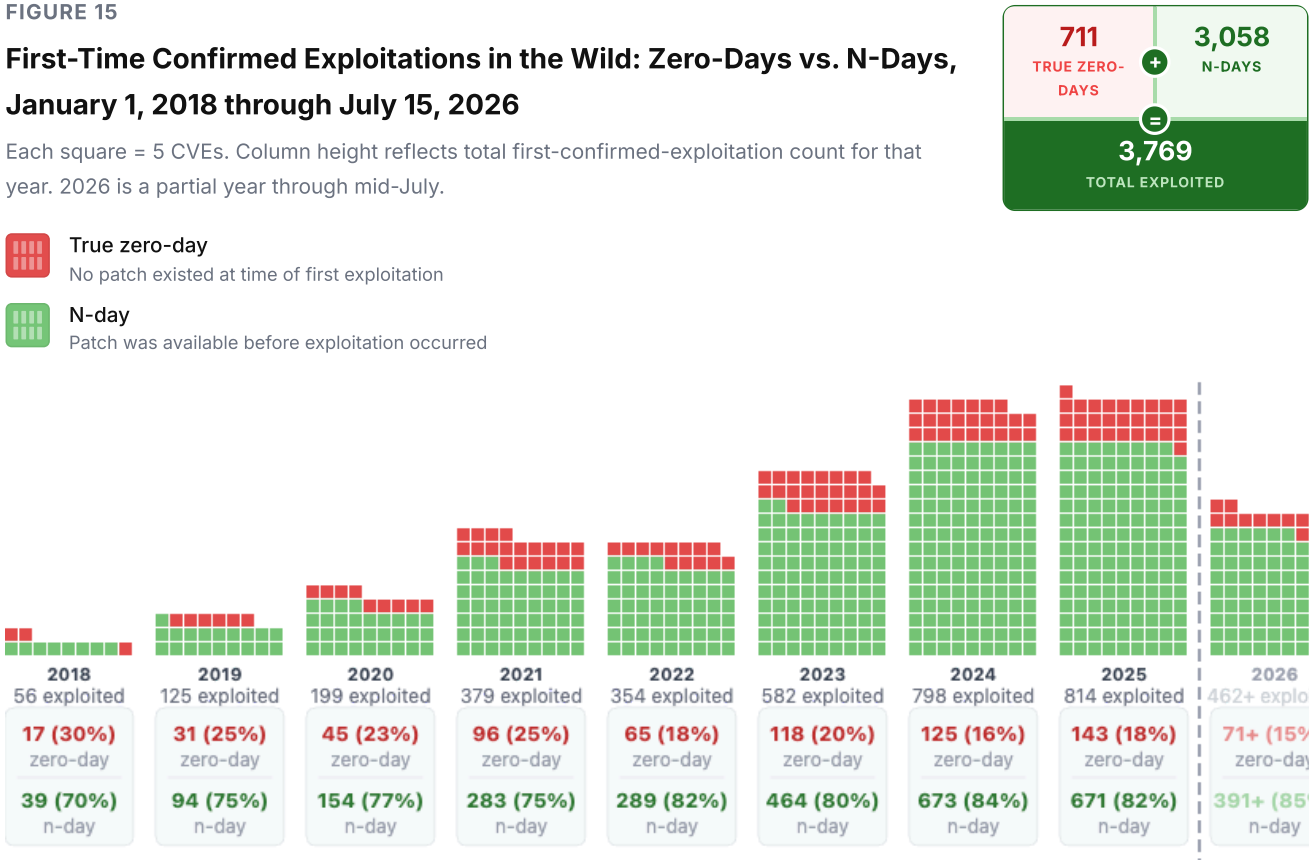
18.9%
true zero-days: no patch existed when first exploited, roughly 1 in 5.

6 of 6
years since 2021, the true zero-day share has stayed between 15% and 26%. See Figure 15.

FIGURE 15

First-Time Confirmed Exploitations in the Wild: Zero-Days vs. N-Days, January 1, 2018 through July 15, 2026

Each square = 5 CVEs. Column height reflects total first-confirmed-exploitation count for that year. 2026 is a partial year through mid-July.



Adversaries have kept that split between 15% and 26% every year since the record began, never trending either way. They exploit more CVEs in total every year, in both categories, but the proportion between the two has stayed put for eight years.

If AI tools were helping adversaries find and exploit zero-days at higher rates, we'd have watched that share climb out of its band by now. We haven't. Adversaries still mostly use CVEs that defenders could have patched, so defenders still decide how most of these stories end.

The 116-day median from Figure 3 says nothing about the adversaries who move fast, and those are the ones security teams really worry about. So in Figure 16, we sorted each year's n-days by speed, from same-day all the way out to 366-plus days, and counted how many adversaries moved in each tier, to find out whether fast exploits are becoming more common.

The share of fast adversaries peaked in 2018

Figure 16 shows adversaries working both ends of the calendar every year, and the fast end is the one shrinking. Adversaries struck within 30 days of the patch on 53.8% of 2018's n-days, then less each year until the share settled around 27% from 2021 through 2023. It jumped to 38.0% in 2024, mostly because catalogers logged the Patchstack batch as same-day exploitation, then fell back to 28.3% in 2025, and in 2026 has reached 33.5% so far.

33.5%
of 2026 n-days exploited within 30 days of patch release

30.4%
of 2026 n-days exploited more than a year after a patch was available

391
n-days confirmed through mid-July 2026, tracking above 2025's pace at the same point in the year

FIGURE 16

N-Day Exploitation Speed: Share of Each Year's N-Days by Time from Patch to First Exploitation, January 1, 2018 through July 15, 2026

Time from patch release to first confirmed exploitation: the window defenders had to act before adversaries arrived.

Lower % — Higher % | Linear scale. Faded = still counting (2025–2026)

	2018	2019	2020	2021	2022	2023	2024	2025	2026
Same-day	20.5% 8	8.5% 8	11.0% 17	4.2% 12	8.3% 24	5.0% 23	18.4% 124	11.9% 80	8.2% 32
1-7d	20.5% 8	22.3% 21	9.7% 15	8.1% 23	6.9% 20	14.0% 65	12.0% 81	9.1% 61	14.8% 58
8-30d	12.8% 5	12.8% 12	17.5% 27	14.8% 42	11.1% 32	8.6% 40	7.6% 51	7.3% 49	10.5% 41
31-90d	23.1% 9	24.5% 23	13.0% 20	13.8% 39	14.9% 43	5.8% 27	9.7% 65	11.0% 74	13.8% 54
91-180d	15.4% 6	8.5% 8	10.4% 16	13.4% 38	7.6% 22	9.5% 44	7.1% 48	9.5% 64	11.0% 43
181-365d	7.7% 3	17.0% 16	10.4% 16	11.7% 33	15.2% 44	9.7% 45	7.3% 49	11.9% 80	11.3% 44
366+d	0.0% 0	6.4% 6	27.9% 43	33.9% 96	36.0% 104	47.4% 220	37.9% 255	39.2% 263	30.4% 119
Total	**39**	**94**	**154**	**283**	**289**	**464**	**673**	**671**	**391**

N-days only (3,058 CVEs, 2018+ CVE cohort, with known patch-to-exploit window; true zero-days excluded). Source: CISA KEV, VulnCheck KEV; patch dates from MSRC, CVE API, OSV, Project Zero.

Adversaries do rack up more fast exploitations each year in raw count, 21 in 2018 against 256 in 2024, but only because they exploit more of everything. The end that genuinely grew is the slow one: adversaries took over a year to exploit 47.4% of what they used in 2023, up from zero percent in 2018, because defenders keep growing the backlog of never-patched systems and adversaries collect from it whenever they get around to it.

A handful of vendors account for the majority of n-day exploitation

Do adversaries spread their work across the whole vendor map, or keep going back to the same names? Figure 17 ranks the top 15 vendors by n-day count, year by year. We ask because the answer tells defenders where their time spent patching buys the most protection.

8 of 9
years where Microsoft ranked #1 by n-day count (Apache edged it in 2023), with 245 CVEs across January 1, 2018 through July 15, 2026

19.8%
of 2021's vendor-attributed n-days traced to Microsoft, its peak year, driven by the ProxyLogon and ProxyShell waves

Top 5
vendors (Microsoft, Apache, Cisco, Apple, Google) account for ~19% of all n-days with known vendor

15
vendors appear consistently across years, indicating adversaries concentrate on a stable set of high-value targets

FIGURE 17

N-Day CVE Concentration by Vendor and Exploitation Year, January 1, 2018 through July 15, 2026

Lower % — Higher % | Linear scale. Faded = still counting (2025–2026)

	2018	2019	2020	2021	2022	2023	2024	2025	2026	Total
Microsoft	15.4% 6	20.2% 19	19.5% 30	19.8% 56	14.9% 43	3.9% 18	4.0% 27	4.0% 27	4.9% 19	245
Apache	2.6% 1	4.3% 4	3.2% 5	6.0% 17	5.2% 15	4.1% 19	2.1% 14	1.3% 9	1.3% 5	89
Cisco	7.7% 3	4.3% 4	3.9% 6	2.5% 7	7.6% 22	0.6% 3	1.0% 7	2.5% 17	2.6% 10	79
Apple	2.6% 1	1.1% 1	0.6% 1	4.2% 12	6.9% 20	1.5% 7	1.0% 7	0.7% 5	3.3% 13	67
Google		4.3% 4	1.3% 2	2.5% 7	2.4% 7	2.4% 11	1.8% 12	1.3% 9	1.3% 5	57
D-Link		3.2% 3	0.6% 1	2.8% 8	1.7% 5	1.7% 8	2.1% 14	1.3% 9	1.3% 5	53
Totolink				0.4% 1	4.8% 14	0.6% 3	1.0% 7	3.0% 20	1.3% 5	50
Oracle	5.1% 2	1.1% 1	3.2% 5	2.1% 6	2.1% 6	1.3% 6	1.5% 10	1.0% 7	1.5% 6	49
VMware	2.6% 1		2.6% 4	3.2% 9	3.1% 9	1.9% 9	1.3% 9	0.6% 4	0.5% 2	47
Adobe	7.7% 3	1.1% 1	1.9% 3	0.7% 2	0.7% 2	2.2% 10	1.0% 7	1.2% 8	0.3% 1	37
Citrix			4.5% 7	1.8% 5	0.7% 2	2.2% 10	0.9% 6	0.7% 5	0.5% 2	37
SonicWall	2.6% 1			1.4% 4	1.0% 3	0.6% 3	0.3% 2	1.6% 11	1.5% 6	30
Ivanti		1.1% 1	1.3% 2	1.8% 5	0.3% 1	0.6% 3	1.3% 9	0.6% 4	1.0% 4	29
Fortinet		3.2% 3		1.4% 4		0.4% 2	0.9% 6	1.2% 8	1.0% 4	27
QNAP			2.6% 4	1.4% 4	1.7% 5	0.2% 1	0.7% 5	0.1% 1	0.3% 1	21
Total	39	94	154	283	289	464	673	671	391	

N-days only (3,058 CVEs, 2018+ CVE cohort). Rows = top 15 software vendors by total n-day count; vendor determined by CPE product configurations, with bundled components (Apache and VMware Spring shipped inside Oracle and other products) reassigned to the originating project rather than the shipping vendor. Columns = year of first confirmed exploitation. Each cell = share of that year's n-days belonging to that vendor. 242 n-days have no identified CPE vendor; they are counted in each year's total but do not appear as a vendor row. 2025 and 2026 are still accumulating. Source: CISA KEV, VulnCheck KEV; vendor attribution from NVD CPE configurations.

Microsoft ranked first every year but one (when Apache edged it out in 2023). Windows, Exchange, Office, and the rest of the Microsoft ecosystem form the biggest single target on the enterprise internet, and when Microsoft patches, hundreds of millions of systems get the update at once, so every adversary knows exactly what to scan for and how long organizations take to apply it. The 2021 peak, 56 n-days and 19.8% of the year's vendor-attributed total, came almost entirely from ProxyLogon and ProxyShell, which drew opportunistic scanners within days of disclosure.

The same 15 vendors have filled the top tier for nine years, and defenders can use that consistency to their advantage. Adversaries know which targets get them access at scale, so they go back. A team that patches Microsoft, Apache, Cisco, Apple, and Google with real discipline covers a significant share of the risk it actually faces.

The same vulnerability classes recur across every year

Vendor rankings tell defenders who gets attacked, but knowing the target doesn't tell you the method: for that, we turn to flaw class, which shows what adversaries use once they pick a victim. Figure 18 ranks the top 15 CWE types across each year's n-days, and it raises an uncomfortable question for the industry:

After decades of secure coding standards, are adversaries still succeeding with the same old flaws?

As it turns out...they are.

#1
OS command injection (CWE-78) with 313 CVEs, the most exploited vulnerability class over the full period, January 1, 2018 through July 15, 2026

CWE-862
Missing Authorization (CWE-862) was absent before 2021 and reached #4 in 2024: the fastest-climbing class in the dataset, driven almost entirely by WordPress plugin exploitation

1 in 3
n-days fall into the top 5 classes: OS command injection, path traversal, SQL injection, cross-site scripting, and unrestricted file upload

15
recurring vulnerability classes appear year after year, showing adversaries exploit the same categories of flaw repeatedly

FIGURE 18

N-Day CVE Concentration by Vulnerability Type and Exploitation Year, January 1, 2018 through July 15, 2026

Lower % — Higher % | Linear scale. Faded = still counting (2025–2026)

	2018	2019	2020	2021	2022	2023	2024	2025	2026	Total
CWE-78 OS Command Injection	5.1% 2	20.2% 19	13.0% 20	13.8% 39	12.8% 37	10.1% 47	8.6% 58	10.1% 68	5.9% 23	313
CWE-22 Path Traversal	7.7% 3	5.3% 5	10.4% 16	6.0% 17	5.9% 17	10.8% 50	6.8% 46	7.3% 49	6.4% 25	228
CWE-89 SQL Injection		3.2% 3	4.5% 7	3.2% 9	1.7% 5	9.9% 46	8.3% 56	7.2% 48	8.7% 34	208
CWE-79 Cross-site Scripting		1.1% 1	2.6% 4	1.1% 3	3.1% 9	4.1% 19	11.0% 74	6.7% 45	3.3% 13	168
CWE-434 Unrestricted File Upload	2.6% 1	2.1% 2	3.2% 5	3.5% 10	2.1% 6	3.2% 15	4.6% 31	3.6% 24	6.4% 25	119
CWE-94 Code Injection	7.7% 3	7.4% 7	1.9% 3	3.5% 10	2.8% 8	3.7% 17	3.4% 23	4.5% 30	4.1% 16	117
CWE-502 Deserialization	5.1% 2	4.3% 4	3.2% 5	2.8% 8	3.8% 11	3.4% 16	3.3% 22	4.6% 31	4.1% 16	115
CWE-862 Missing Authorization			0.6% 1	0.7% 2	1.0% 3	5.2% 24	7.1% 48	2.2% 15	3.6% 14	107
CWE-787 Out-of-bounds Write	12.8% 5	10.6% 10	7.1% 11	6.7% 19	6.6% 19	1.5% 7	1.8% 12	2.2% 15	1.3% 5	103
CWE-77 Command Injection			0.6% 1	1.1% 3	2.8% 8	3.4% 16	3.6% 24	4.3% 29	3.6% 14	95
CWE-306 Missing Auth (Critical)		2.1% 2	2.6% 4	2.8% 8	1.7% 5	3.2% 15	1.6% 11	2.7% 18	5.1% 20	83
CWE-287 Improper Authentication	5.1% 2	2.1% 2	2.6% 4	3.2% 9	2.4% 7	2.6% 12	2.7% 18	3.0% 20	1.5% 6	80
CWE-20 Improper Input Validation	12.8% 5	3.2% 3	3.9% 6	1.1% 3	4.5% 13	2.2% 10	1.3% 9	1.6% 11	2.3% 9	69
CWE-918 SSRF		2.1% 2	0.6% 1	3.2% 9	1.4% 4	2.4% 11	1.9% 13	2.8% 19	2.6% 10	69
CWE-200 Information Exposure		1.1% 1		1.8% 5	1.7% 5	2.6% 12	1.3% 9	2.5% 17	2.6% 10	59
Total	**39**	**94**	**154**	**283**	**289**	**464**	**673**	**671**	**391**	

N-days only (3,058 CVEs, 2018+ CVE cohort). Rows = top 15 CWE types by total n-day count. Columns = year of first confirmed exploitation. Each cell = share of that year's n-days with that CWE. 249 n-days have no specific CWE; they are counted in each year's total but do not appear as a row. 2025 and 2026 are still accumulating. Source: CISA KEV, VulnCheck KEV; CWE from NVD bulk feeds and MITRE CVE API.

OS command injection led the full period with 313 CVEs, with path traversal and SQL injection close behind. The order at the top shuffles from year to year (adversaries exploited more path traversal than anything else in 2023, more XSS in 2024, more SQL injection in 2026) but the same handful of classes fills the top spots every time. Every secure coding

standard warns against them, every security training covers them, and adversaries keep exploiting them anyway, which says less about adversary sophistication than about how slowly the industry closes gaps it has known about for decades.

The bad guys aren't using exotic flaws. They didn't have to.

N-day exploitation is split nearly evenly between commercial and open-source software

So far we've answered who adversaries target and which flaws they exploit. We still need to know where they aim by software category, because defenders who manage mixed environments often patch commercial software on one schedule and open source on another.

The instinct makes some sense, since a Microsoft patch and an npm package update reach production by very different roads. But nothing we found says defenders should treat either category as the safer one. Figure 19 shows how adversaries split their n-day exploitation between the two, year by year.

50/50
OSS and commercial n-days are nearly equal across January 1, 2018 through July 15, 2026 (1,549 vs. 1,498 CVEs)

2023
the year OSS n-days (255) first surpassed commercial (209) and has grown faster every year since

23×
growth in OSS n-day count from 2018 (17) to 2024 (394), outpacing commercial growth over the same period (14×)

60%
of 2026 n-days traced to open-source software through mid-July, up from 36% in 2022 as WordPress plugin exploitation drove a sharp OSS surge from 2023 onward

FIGURE 19

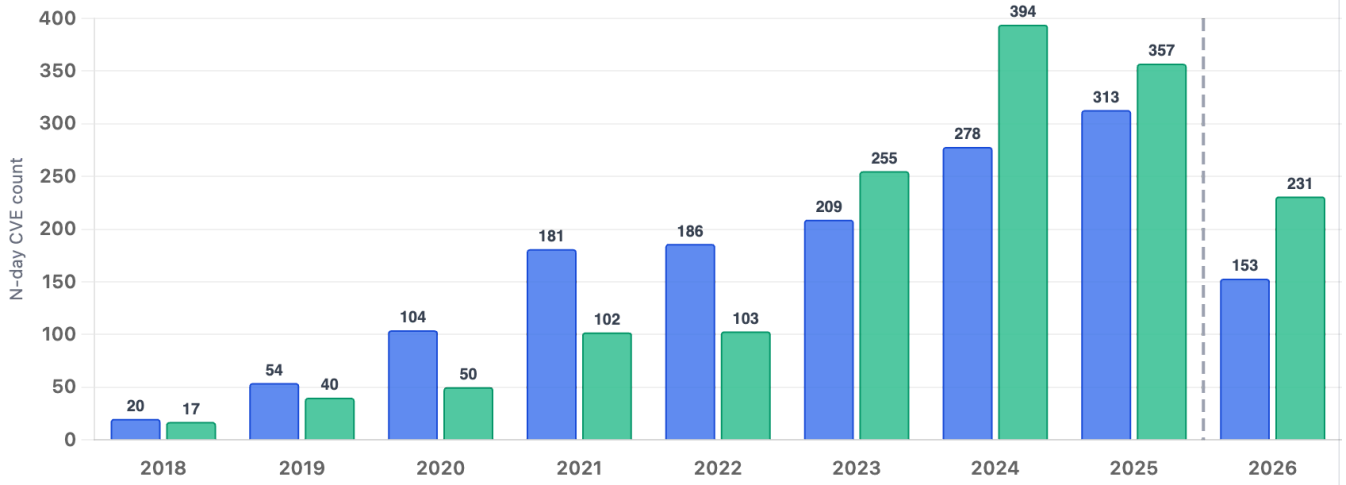
N-Day CVEs by Software Category and Exploitation Year, January 1, 2018 through July 15, 2026

Commercial
Closed-source vendor products (is_oss = false)

All OSS
Open-source software including WordPress plugins (is_oss = true)

THE SHIFT

OSS n-days first surpassed commercial in 2023 and grew 23x from 2018 to 2024. By 2026, 60% of confirmed n-days trace to open-source software.



N-days only (3,047 CVEs in the 2018+ cohort with known OSS classification; 11 with unknown classification excluded). Grouped bars show Commercial (blue) and All OSS (green) counts by year of first confirmed exploitation. Labels = share of that year's n-days. 2025 and 2026 are still accumulating. Source: CISA KEV, VulnCheck KEV; OSS classification from CNA rules, OSV, NVD CPE, package registries, and web search.

Across the full period, adversaries exploited 1,549 open-source n-days and 1,498 commercial, a nearly even split, but they didn't split their work evenly in any single year.

Through 2022 they favored commercial software, taking about 65% of their n-days there in 2020 and 2021. Then they flipped, for two reasons: organizations had adopted enough open source to make its flaws worth exploiting at scale, and Wordfence and Patchstack began cataloging exploited WordPress plugin CVEs in volume, often recording many on a single date. From 2018 to 2024, adversaries multiplied their open-source n-day count 23 times over, vs. 14 times for commercial.

The patch-to-exploit timeline differs between commercial and open-source software

Figure 19 told us where adversaries went. Now we need to know how fast they moved once the patch shipped, so in Figure 20 we compared adversary timelines on open-source and commercial software. If adversaries exploit one category faster than the other, defenders shouldn't patch the two on the same schedule.

91d
OSS median defender window in 2026 (through mid-July), up from 66d in 2024 as the WordPress same-day batch effect fades

119d
commercial median defender window in 2026, down from 232d in 2025

1.3×
faster OSS exploitation than commercial in 2026, down from 3.1× in 2024: the two categories are converging

Both <150d
in 2026, commercial (119d) and OSS (91d) windows sit closest together in the series, under similar time pressure

FIGURE 20

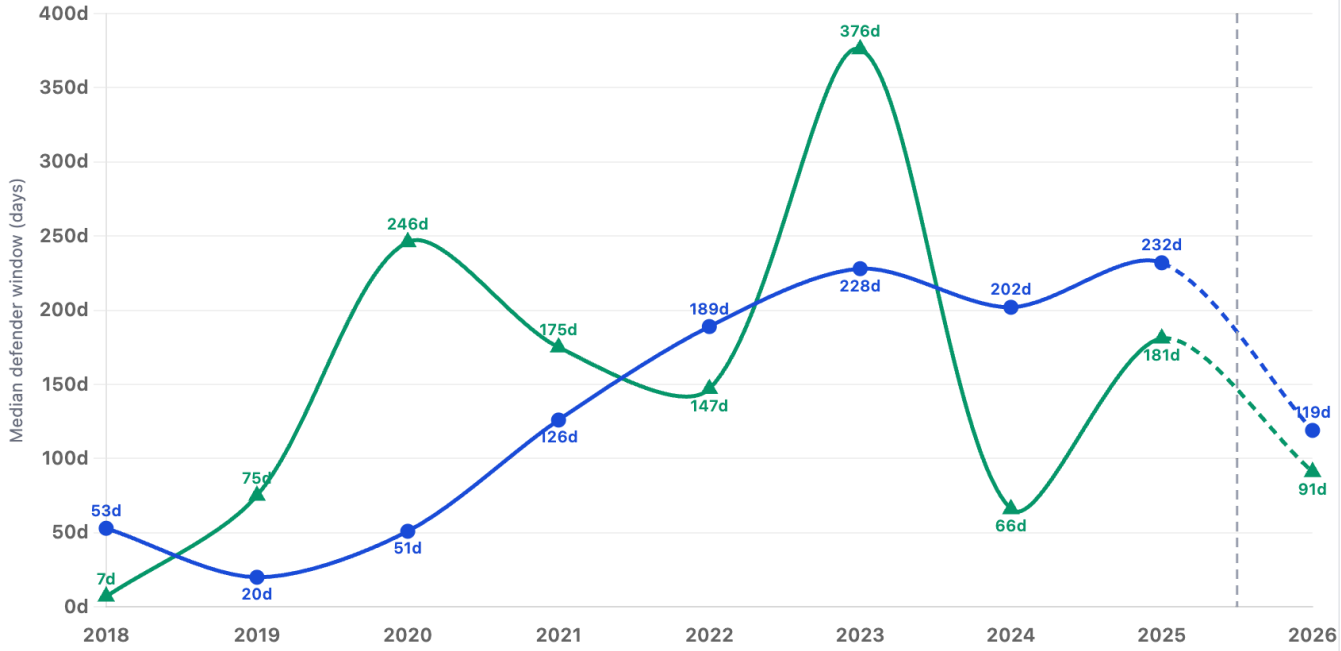
N-Days: Median Days from Patch to First Exploitation, Commercial vs. All OSS, by Exploitation Year, January 1, 2018 through July 15, 2026

Commercial
Median days from patch release to first exploitation, commercial CVEs

All OSS
Median days from patch release to first exploitation, OSS CVEs

CONVERGENCE

Commercial and OSS windows converged in 2026, both near 100 days (119 and 91). The 3.1x OSS speed advantage in 2024 has shrunk to 1.3x.



N-day CVEs only (exploitation date minus patch date, in days). Grouped by year of first confirmed exploitation. Commercial = closed-source (1,498 CVEs); All OSS = open-source including WordPress (1,549 CVEs); 11 CVEs with unknown classification excluded. 2025 and 2026 are still accumulating. Source: CISA KEV, VulnCheck KEV.

The two categories split apart in 2024. Adversaries kept exploiting commercial software on their usual timeline, a 202-day median, but they cut the open-source median to 66 days, from 376 the year before. The WordPress wave explains most of that drop: catalogers recorded over 100 Patchstack CVEs as exploited the same day the patch shipped, which dragged the open-source median down.

Even discounting that artifact, adversaries really *have* sped up against open source, because researchers and automated scanners now spot freshly patched projects quickly and test them against public install bases where every version number is visible.

Through 2024, then, teams running mostly commercial software had more time than the median suggested. In 2026 that cushion thinned: adversaries took both categories under 150 days for the first time: 119 for commercial and 91 for open source. We're watching that convergence closely.

2026 N-Day Snapshot: Who, What, and How Fast

CISA and VulnCheck logged 391 exploited n-days through mid-July 2026, more than they had logged by the same point in 2025. Adversaries took a median of 116 days after the patch to show up, but they didn't cluster around that median: they hit 33.5% of these CVEs within 30 days and waited more than a year on 30.4%. Security teams can't plan around a median when a third of attacks come months sooner.

391
n-days first exploited in 2026 (through mid-July)

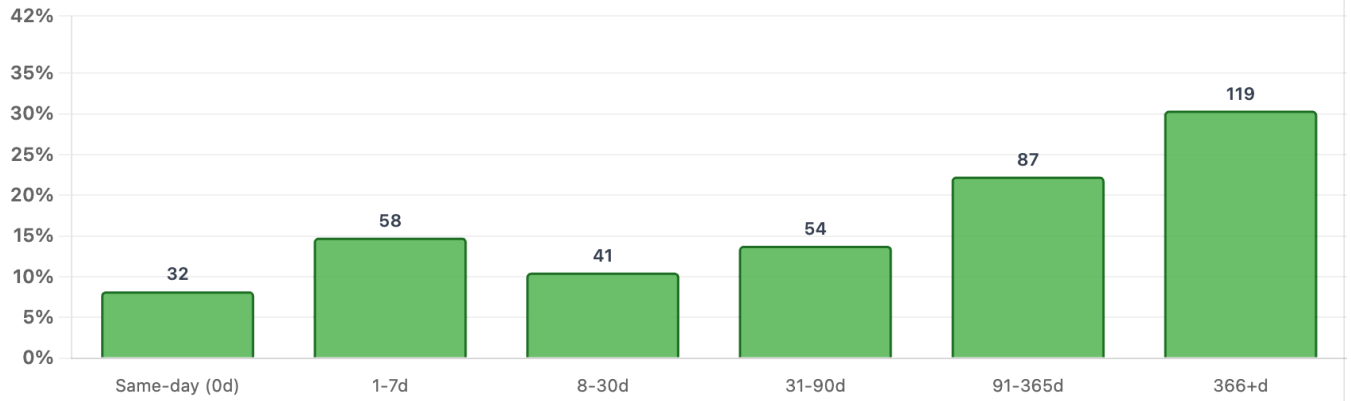
116 days
median time from patch release to first exploitation

33.5%
exploited within 30 days of patch release

30.4%
exploited more than a year after a patch was available

FIGURE 21

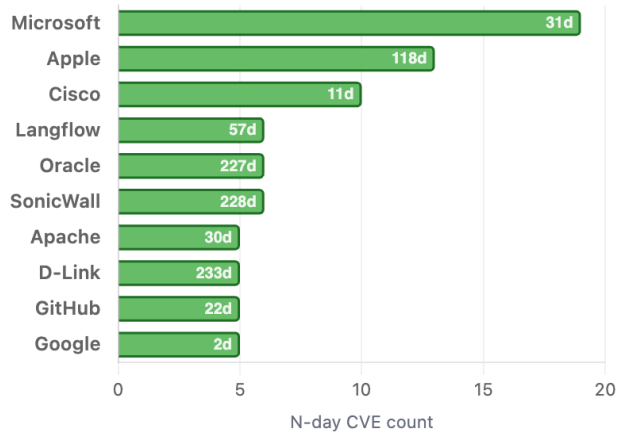
2026 N-Day Exploitation Speed: Time from Patch Release to First Known Attack



391 n-days with a recorded patch date, first exploited January–July 2026. Bars show share of CVEs in each time-from-patch bucket. Source: CISA KEV, VulnCheck KEV; patch dates from MSRC, CVE API, and vendor advisories.

FIGURE 22

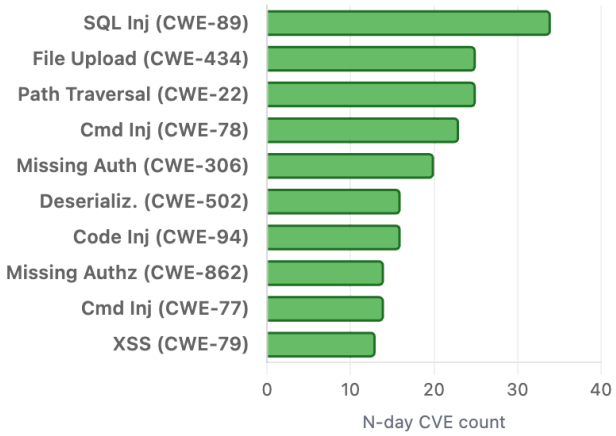
Top 10 Vendors by 2026 N-Day Count



Vendor from NVD CPE configuration. Numbers in bars = median days from patch to first exploitation. 85 n-days (21.7%) had no CPE vendor recorded and are excluded.

FIGURE 23

Top 10 Vulnerability Types by 2026 N-Day Count



CWE from NVD bulk feeds. Percentage = share of 2026 n-days with that CWE. Source: CISA KEV, VulnCheck KEV.

Our vendor and flaw data explain who those fast adversaries went after. In 2026, they exploited Cisco CVEs at a lightning-fast 11-day median, with Microsoft at 31 and Apache at 30, while leaving Oracle, SonicWall, and D-Link sitting patched-but-unexploited for 225 to 235 days.

Langflow, an open-source AI workflow platform, recently entered the top 10 for the first time, 6 CVEs at a 57-day median, which tells us adversaries have started treating AI-adjacent tooling as a real target. Their favorite flaw classes (SQL injection, path traversal, and command injection) are exactly the ones with mature public exploit tooling and broad scanner coverage.

Key takeaway: adversaries move fast where the tooling exists and the target is popular, and slow everywhere else.

Security teams can use that. A "patch critical CVEs within 30 days" policy fits Microsoft and Cisco estates, where adversaries actually move that fast, but does nothing useful for Oracle middleware or unmanaged IoT hardware, where adversaries take seven months. CVSS scores won't make that distinction: adversaries treat a 9.8 on a Cisco router and a 9.8 on a D-Link device twenty times differently. Teams that check a CVE's vendor and flaw class against how fast adversaries historically moved on that combination can set patching deadlines that match reality. Teams that rank by severity alone are missing the way risk actually accrues.

FIGURE 24

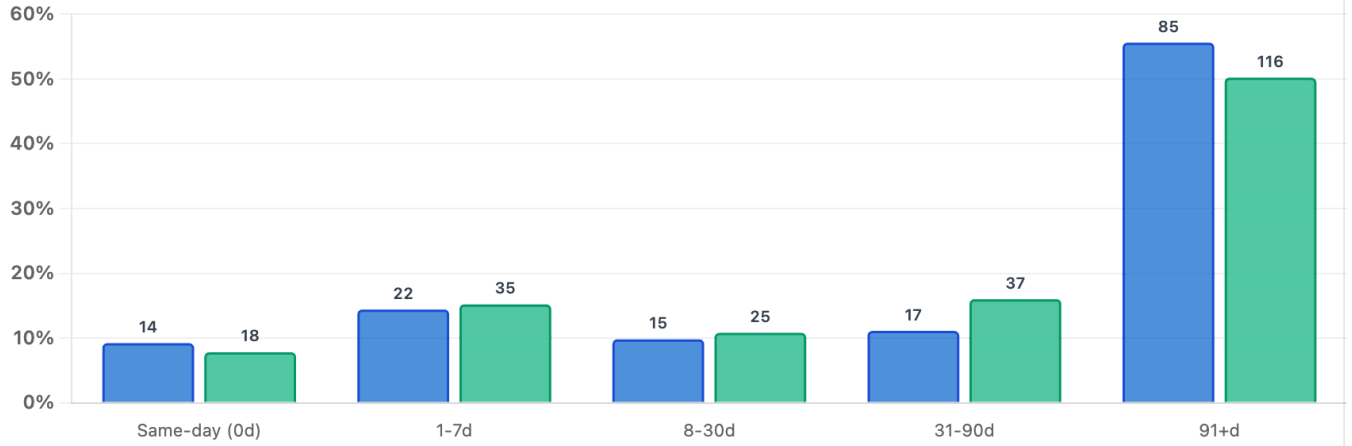
2026 N-Days: Exploitation Speed by Software Category

Commercial
n=153 CVEs. Median defender window: 119 days

All OSS
n=231 CVEs. Median defender window: 91 days

2026 (PARTIAL, MID-JULY)

60% of 2026 n-days are OSS. Bars show each category's share of CVEs by exploitation speed tier, not raw counts.



2026 n-days only (first exploited January–July 2026). Bars show each category's share of CVEs by time from patch availability to first known exploitation. 7 CVEs with unknown OSS classification excluded. Commercial median 119d vs. All OSS median 91d.

If AI were making adversaries broadly faster, we'd see them crowding the same-day and sub-7-day rows of this last chart and abandoning the year-plus row.

They're doing neither.

They still exploit the same flaw classes they've led with for a decade, they still take over a year to exploit three of every ten n-days, and they still strike within 30 days about a third of the time, the same third as the past five years. Whatever AI eventually does for adversaries, through mid-2026 they haven't gotten faster.

The Bottom Line for N-Days

We've said it several ways across these pages, so here is the shortest one: **uninstalled patches have caused far more damage than unpatched flaws ever have**. Adversaries exploited four CVEs with an existing fix for every one without, 81.1% of the dataset, nine straight years of a similar ratio.

The rest of what we found follows from the two adversary crowds. Against same-day adversaries, teams need compensating controls running before the advisory publishes. Against the sub-30-day crowd, they need a deadline they actually hit. Against everything slower, they need selection, because nobody's patching throughput covers the full NVD feed, so somebody has to decide which CVEs the bad guys will actually come for.

This kind of selection works, because the bad guys repeat themselves. You've now read the evidence: Microsoft on top in eight of nine years, the same 15 vendors in the top tier every year, OS command injection, path traversal, and SQL injection leading since 2018.

Security teams that focus their patching on those vendors and flaw classes cover a massive share of the risk they face, and if researchers keep growing the catalog while adversaries keep exploiting the same slice of it, that kind of focus becomes the only strategy that can scale.

How much time do defenders realistically have?

Between 2018 and 2025, adversaries waited anywhere from a median of 24 days to a median of 327 after the patch, and so far in 2026 they're waiting 116. No security team should set a deadline off those medians, though, because a third of 2026's attacks came inside 30 days and three in ten came after a year. The useful question is which of those groups will come for a given CVE, and a CVE's vendor and flaw class answer that question better than its CVSS score does.

Does the window vary predictably by software category or vulnerability type?

Yes, by margins big enough to change how a team operates. In 2023, adversaries exploited Missing Authorization CVEs within 30 days of the patch 54% of the time, and SQL injection just 17%. Vendor differences run even wider: across the full period, adversaries went after Google and Adobe n-days within 30 days more than half the time, 67% and 51%, while they gave Oracle seven months and the IoT vendors D-Link and Totolink years, striking within 30 days on just 8% and 4% of their CVEs. Then in 2026, for the first time, they exploited commercial and open-source software at nearly the same speed, a 119-day median against 91. Security teams can't use CVSS scores to read the risk picture here: vendors and vuln classes matter much more.

Is adversary speed increasing, and what does the trend suggest about AI's role?

Adversaries have kept their zero-day share between 15% and 26% every year since 2021, and they took their biggest zero-day swings back in the browser-and-mobile years, not recently. What they have changed is the slow end: in 2018 they exploited nothing more than a year after its patch, and by 2023 they took that long on 47% of their n-days, because security teams keep abandoning systems and leaving them permanently unpatched. Once we correct for NVD's lag, adversaries show no speedup at all. If AI-assisted acceleration ever arrives, adversaries will show it first in the same-day and sub-7-day rows of Figure 16, and they haven't yet.

Conclusion: Reports of the Vulnpocalypse Have Been Greatly Exaggerated

The Vulnpocalypse was supposed to be here by now.

Pundits promised AI-armed adversaries weaponizing every fresh CVE within hours, and security teams drowning shortly after.

We spent this report checking that forecast against eight and a half years of confirmed exploitation, and discovered that the adversaries keep contradicting it. They exploit a tiny (and growing tinier) share of what researchers publish, and they almost never strike before a patch exists. When they use an n-day, they wait longer now than they did in 2018.

AI has changed one group so far, and we can name it: researchers.

They're publishing CVEs faster than at any point in history, and every rising trend in this report traces back to them. The adversaries reading their output stayed picky, stayed slow, and stayed loyal to the same vendors and the same old flaws. That's why we can predict with some assurance that the noise will keep growing, but the signal won't.

Security teams that accept this simple fact can do less while preventing more: patch what adversaries use, learn which crowd is coming for a given CVE, and let the people counting CVEs count in peace. Teams that chase the catalog will work harder every year to stop the same small set of attacks.

We're not promising adversaries will stay slow. The catalogs run years behind, the strongest AI tools are barely two years old, and we've named the rows in Figure 16 where acceleration would show first.

We'll keep running the numbers. If adversaries change, we'll say so before the pundits do.

Until then, the count stands at thevulnpocalypse.com: 147 days in, civilization intact.

Data Methodology

Author: Jeremiah Grossman

Data captured: July 15, 2026

CONTENTS

1. Purpose

This document explains the data collection and classification methodology behind two reports, *True Zero-Days: What the Data Actually Shows* and *N-Days: The Defender Window*. Both draw from the same set of 3,769 classified exploited CVEs from 2018 through 2026. The purpose is transparency. The sources, decision rules, and edge cases behind every figure are documented so the findings can be understood and checked against the public record.

True Zero-Days examines CVEs where exploitation was recorded before a patch existed. *N-Days: The Defender Window* examines CVEs exploited after a patch was available and measures how much time defenders had. One question determines which report a CVE belongs to. Did a patch exist before exploitation was recorded?

CLASSIFICATION	CONDITION	COUNT (2018-2026)	RATE
`True zero-day`	Exploit before patch	**711**	**18.9%**
`N-day`	Exploit after patch	3,058	81.1%
Total classified		**3,769**	**100%**

The dataset holds 3,771 confirmed-exploited CVEs. Two (CVE-2018-20841 and CVE-2021-0027) have no patch date available and cannot be classified, so the 3,769 analyzed throughout these reports exclude them. Across the full 2010–2026 dataset, 10 such cases exist, all legacy or end-of-life products for which no dated fix could be found or none was ever released (for example, the retired Apache Continuum project and the WIFICAM/GoAhead OEM cameras). Their existence is recorded here but they are not counted.

We use "true" in "true zero-day," rather than just "zero-day," to distinguish our metric from the Zero Day Clock (ZDC), which counts a CVE as a zero-day when exploitation is detected on or before the National Vulnerability Database (NVD) publishes the record. Because NVD has recently averaged 25 days behind the vendor's patch, that definition sweeps in

CVEs that already had a fix. The full comparison and its effect on the numbers are in *True Zero-Days*. This analysis defines a zero-day the way the field always has, exploitation before a patch exists. Everything else is an n-day, and the next question is how large n was.

2. Data Sources

Exploit Signal Sources

For each CVE, the exploit date is the earliest date on which in-the-wild exploitation was recorded. Two catalogs are used. The earlier date across both is taken.

CISA Known Exploited Vulnerabilities (KEV): A public catalog of vulnerabilities with confirmed exploitation. The `dateAdded` field records when CISA added the entry. When CISA launched in November 2021, it added historical entries without recording when exploitation actually occurred, so the 31 CVEs dated 2021-11-03 carry unreliable exploitation dates and are set aside in any pre-2022 timing analysis.

VulnCheck KEV: A larger catalog that includes CISA KEV as a subset. Only the `date_added` field and the dates within `vulncheck_reported_exploitation` are used. Each record also has an XDB section, which tracks when proof-of-concept exploit code was added to VulnCheck's repository, not when attacks occurred in the wild; those XDB dates are excluded.

NVD Publication Dates

For each CVE, the NVD publication date is the date that CVE record first appeared in the National Vulnerability Database. This is fetched from the NVD API. If NVD lacks a record (possible during processing backlogs), the MITRE CVE API provides a fallback date.

Patch Date Sources

The patch date is the earliest date on which a fix was publicly available. Multiple sources are queried and the earliest date across all of them wins. Using the earliest possible date gives defenders the maximum credit for patch availability, making the true zero-day findings conservative.

SOURCE	COVERAGE (2018-2026)	NOTES
MSRC (Microsoft Security Response Center)	~12% (466 CVEs)	Patch Tuesday bulletin release dates. Reliable to the calendar day and the highest-confidence source for Microsoft CVEs. Accessible via MSRC's public API.
CVE API advisory dates	~67% (2,514 CVEs)	Each CVE record in the MITRE CVE API carries two dates: `datePublic`, which is when the vendor or reporter made the vulnerability publicly known (typically the advisory or patch release date), and `datePublished`, which is when the CNA formally filed the CVE record with MITRE. These can differ by days, weeks, or months. `datePublic` is the closer proxy for patch availability; `datePublished` reflects administrative processing lag. The earlier of the two is used. For vendors who self-assign CVEs and publish them the same day they release a fix, the two dates coincide. For vulnerabilities assigned by MITRE as CNA of last resort, `datePublished` may significantly lag the actual fix date, making it a less reliable proxy. This source covers non-Microsoft CVEs and serves as fallback for all others.
GitHub (commits, release tags, PRs, GHSA)	~10% (360 CVEs)	GitHub commit history, release tags, pull requests, and GitHub Security Advisories. Used when a referenced commit or release provides an earlier date than the CVE API. Applied via automated enrichment pass after initial collection.
Patchstack database	~3% (123 CVEs)	For CVEs assigned by Patchstack as CNA, the Patchstack DB publication date equals patch availability: Patchstack only publishes after the vendor has shipped a fix. Applied as a batch correction to all Patchstack-CNA CVEs.
Google Project Zero 0days-in-the-wild spreadsheet	~3% (112 CVEs)	Project Zero manually tracks vendor advisory dates for Chrome, Apple, Mozilla, ARM, Fortinet, and others. Their `Date Patched` is used when it is earlier than the CVE API date. Available publicly from the Project Zero website.
Apple HT advisory pages	~1% (33 CVEs)	Apple publishes security updates to HT advisory pages before formally publishing CVE records to NVD. The advisory's "Released" date is used when it predates the CVE API date.
NVD reference URLs	~1%	Vendor advisory URLs tagged in NVD bulk data as "Vendor Advisory" are fetched and parsed for release dates earlier than the CVE API date.
MITRE CVE API reference URLs	~1%	Reference URLs embedded in CVE records via the MITRE CVE API. Fetched and parsed for advisory or changelog dates earlier than the current patch date.
OSV / package registries	<1%	Open Source Vulnerabilities database and package registries (PyPI, npm, RubyGems, WordPress.org changelog API). Used for OSS packages where a version release date predates the CVE API date.
Named vendor advisory pages	<1%	Structured advisory pages from JVN (Japan Vulnerability Notes), Mozilla, CISA, Zyxel, IBM, and others. Fetched during a targeted enrichment pass against CVEs with named vendor CPEs.
Wayback CDX	<1%	Internet Archive CDX API, used to find the earliest archived version of a vendor advisory page. Applied with a year-sanity filter and a 180-day cap to avoid false positives from domain-level crawl dates.
Manual vendor verification	<1%	Individual cases where automated enrichment found no earlier date. Vendor changelogs, release notes, blog posts, and security advisories were checked directly. Sources recorded individually by CVE (e.g., `litespeed_blog`, `gfi_release_notes`, `bricks_changelog`, `cisco_psirt`).

CHROME PATCH DATES

Microsoft's Patch Tuesday bulletins sometimes include Chrome CVEs (because Edge is Chromium-based), and Edge and Chrome release on different schedules. For Chrome CVEs, the correct patch date is the Chrome stable channel release date from the Project Zero spreadsheet, not an Edge Patch Tuesday date.

3. Data Collection Pipeline

The full dataset is assembled in four logical stages before classification is applied.

Stage 1: Build the exploit signal set. For each CVE in CISA KEV and VulnCheck KEV, record the earliest exploitation date across both catalogs. XDB dates are excluded at this stage. The result is one exploitation date per CVE.

Stage 2: Fetch NVD publication dates. For each CVE, retrieve its NVD publication date. Drop CVEs with no NVD date. Also drop CVEs with exploit dates before 2010-01-01 (outside the analysis period) and CVEs where the exploit predates NVD publication by more than 180 days (implausible outliers).

Stage 3: Collect initial patch dates. For each remaining CVE, query MSRC, the CVE API, and the Project Zero spreadsheet. Record the minimum date found across all sources as the authoritative patch date, along with which source produced it. At this stage roughly 85% of CVEs have a patch date from the CVE API and 12% from MSRC, with Project Zero covering ~3%.

Stage 4: Enrich patch dates. Initial collection uses proxy dates for many CVEs. The CVE API's `datePublished` reflects when the CNA filed the record, not necessarily when the vendor released the fix. Multiple enrichment passes improve these dates by finding earlier, more accurate sources. Passes run in sequence. Each earlier date found supersedes the current value.

The enrichment passes, in order: Apple HT advisory pages (for Apple CNA CVEs); Chrome stable channel correction via Project Zero (for Chrome CVEs with MSRC dates from Edge Patch Tuesday); NVD reference URL extraction (vendor advisory URLs in NVD bulk data); GitHub commits, release tags, PRs, and GHSA; OSV API and package registries (PyPI, npm, RubyGems, WordPress.org changelog API); MITRE CVE API reference URLs; VulnCheck KEV reference URLs; named vendor advisory pages (JVN, Mozilla, CISA, Zyxel, IBM, and others); Wayback CDX with year-sanity filter; web search (DuckDuckGo) for obscure vendors with no indexed advisory URL; Patchstack database batch correction for all Patchstack-CNA CVEs; Wordfence timeline "Disclosed" dates for retroactive cataloging cases; and manual vendor verification for individual high-confidence cases.

4. Computed Fields and Chart Inputs

Once the four core data points are collected (exploit date, NVD publication date, patch date, and CVE year), the following fields are computed. Each one feeds one or more of the published charts.

Defender window. Exploitation date minus patch date, in days. Negative means the exploit predated the patch, which by definition makes it a true zero-day. Positive means the defender had that many days to apply the patch before exploitation occurred. The defender window charts answer a single question. Of the CVEs where a patch existed before exploitation, how quickly did adversaries act?

Zero-days vs. n-days. A zero-day is a CVE exploited before a patch existed (negative defender window). An n-day is a CVE exploited after a patch was available, where "N" is the number of days since patch release. In the dataset, there are 711 zero-days and 3,058 n-days. The zero-day charts analyze the 711. The defender window charts analyze the 3,058.

ZDC rate. For comparison with the Zero Day Clock metric, the ZDC rate is the share of CVEs where exploitation was recorded on or before the date NVD published the record, a time-to-exploit (TTE) of zero or less. In the 2018-2026 dataset, that is 1,243 CVEs, or 33.0%. This is the figure ZeroDayClock reports as its "zero-day" rate. Of those 1,243, only 711 (18.9%) are true zero-days where no patch existed at time of exploitation. The other 532 CVEs had patches available before exploitation was recorded. 327 were exploited after patch release but before NVD publication (

`post_patch_pre_nvd`, 8.7%), and 205 were exploited on the same calendar day NVD published but after a patch was already available. The 14.1-percentage-point gap between ZDC's 33.0% and the true zero-day rate of 18.9% is the NVD inflation effect.

Vendor attribution. Vendor names are resolved in two phases. Phase 1 extracts the vendor from NVD bulk CPE configurations, preferring application CPEs marked as vulnerable over hardware or OS CPEs. Phase 2 applies a CNA email domain fallback for CVEs with no CPE data, primarily MITRE-assigned and VulnCheck-assigned CVEs where NVD has no CPE configuration. Combined coverage reaches 86.5% of CVEs with a named vendor. The remaining 13.6% is concentrated in VulnCheck-assigned IoT and embedded device CVEs and pre-2018 CVEs. The CNA (the organization that filed the CVE) is tracked separately and used for CNA-level analysis. It is not the same as the product vendor for MITRE- and VulnCheck-assigned CVEs.

Open source vs. commercial classification. Each CVE is classified as open source (OSS) or commercial software across six phases. Phase 1 applies CNA identity rules. CNAs with known OSS scope (Jenkins, Eclipse, Apache, etc.) are classified OSS, and CNAs with known commercial scope (Juniper, Samsung, IBM, VDE, etc.) are classified commercial. Phase 2 queries the Open Source Vulnerabilities (OSV) database, which only indexes OSS packages, to confirm additional OSS CVEs. Phase 3 uses NVD CPE vendor and product strings to match against known OSS and commercial product lists. Phase 4 checks package registries (PyPI, npm, RubyGems, WordPress.org) for package name matches and parses NVD reference URLs for OSS-indicator patterns. Phase 5 scans NVD description text for OSS keywords and product names, and searches GitHub for repository matches. Phase 6 applies named-product description patterns and, for remaining unknowns, a DuckDuckGo web search. A validation pass then cross-checks all classified CVEs against description patterns, correcting 19 false positives where weak sources had misclassified commercial products as OSS. Final coverage is OSS 50.6% (1,908 CVEs), commercial 49.1% (1,852 CVEs), and unknown less than 1% (11 CVEs).

Vulnerability class (CWE). CWE (Common Weakness Enumeration) identifiers are collected in two phases. Phase 1 retrieves CWEs from NVD bulk feeds for the 2018-2026 range, covering 91.6% of 2018-2026 CVEs with a specific CWE identifier. Phase 2 queries the MITRE CVE API for CVEs returning `NVD-CWE-noinfo`, `NVD-CWE-Other`, or no CWE, upgrading 51 additional CVEs to specific CWEs. Final coverage for 2018-2026 is 91.6% specific CWE, 7.5% noinfo/Other, and 0.9% no CWE. For the full dataset including pre-2018 CVEs, specific CWE coverage is 76.6%, with the gap concentrated in pre-2018 CVEs where NVD records are sparse. CWE values are used as provided. No manual reclassification is applied.

Total CVE volume. Charts that express the exploited population as a share of all published CVEs (such as the 0.28% figure for true zero-days) require a separate count of all CVEs in NVD for each year, not just the exploited subset. This count is fetched from the NVD API using year-based queries and is not part of the main dataset.

Once all three dates are collected and these fields are computed, the initial bucket assignment follows from comparing them. The classification methods in Section 5 then refine the bucket for same-day cases and WordPress CVEs.

Analysis window: The dataset covers CVEs from January 1, 2018 through July 15, 2026, where "year" refers to the year in the CVE ID (CVE-2021-NNNNN = 2021), not the year exploitation was recorded. This CVE-ID year governs only which CVEs enter the 2018+ analysis cohort. The year-by-year charts in both reports group CVEs by the year of first confirmed exploitation, so per-year chart totals can differ from CVE-ID-year counts. CVEs before 2018 exist in the raw collection but are excluded from both published analyses due to lower data quality from CISA KEV backfill concentration.

ENRICHMENT COVERAGE NOTE

The OSS classification, CWE enrichment, and vendor attribution passes were all completed before a final incremental KEV refresh that added 73 CVEs to the full dataset (66 in the 2018-2026 dataset). The counts cited above for those three enrichments reflect the pre-refresh set. The newly added CVEs received initial classification during collection where data was available but did not go through the full multi-phase enrichment pipeline. The effect on reported rates is negligible given the small count of affected CVEs.

5. Classification Methods

DEFINITION

True zero-day: A vulnerability that was actively exploited in the wild before a patch was available to defenders.

In practice: A CVE is counted as a true zero-day if it satisfies at least one of the three criteria below. Each criterion requires positive evidence of pre-patch exploitation. When evidence is absent or ambiguous, the CVE is not reclassified. The reported rate is therefore a lower bound.

Method 1: Date-Math (Primary)

MECHANISM

Exploit date is strictly earlier than patch date.

When the earliest recorded exploitation signal predates the earliest available patch date by at least one calendar day, the CVE is classified as `true_zero_day`. This is the primary classification mechanism and handles the majority of true zero-days, where the gap between exploitation and patch is large enough to be unambiguous.

Limitations: All dates are calendar-day precision, so when the exploit date and patch date are the same day, the intra-day sequence cannot be determined. Those cases are handled by Method 2. Exploitation dates are also proxies, not ground truth. They reflect when a catalog recorded the event, not when exploitation began.

Method 2: Coordinated-Disclosure Acknowledgment

MECHANISM

Exploit date equals patch date, and the vendor's own advisory explicitly acknowledges active exploitation at time of patch release.

Many same-day cases are coordinated disclosures. The vendor detected active exploitation, built a patch, and released it with notification to CISA on the same calendar day. The actual exploitation preceded the patch by an unknown margin, but the dates in the catalogs are identical. All 222 same-day CVEs from 2018 onward were investigated through a five-tier hierarchy. Each tier handled cases the prior tier could not resolve.

Tier 1: NVD description text (34 reclassified). The NVD description is searched for date-of-exploitation language ("exploited in the wild," "actively exploited before") and explicit pre-patch date references.

Tier 2: External article URL (10 reclassified). Reference URLs in the CVE record pointing to external articles are fetched and checked for zero-day terminology in the title or body text.

Tiers 3 and 4: Pattern variants (17 reclassified). The additional NVD description patterns include month-and-year phrasing implying pre-patch exploitation, year-only references earlier than the NVD publication year, and CVE-specific zero-day confirmation extracted from fetched article content.

Tier 5: Advisory content fetch (24 reclassified). Full advisory page text is fetched from an expanded domain list (Fortinet blog, Talos, Google/Mandiant blog, VulnCheck, SANS ISC, horizon3.ai) and searched for strong pre-patch exploitation language. Advisory pages from Cisco PSIRT, Fortiguard, Broadcom, and Ivanti are also checked for explicit exploitation acknowledgment.

In total, 85 of the 222 same-day CVEs were reclassified to `true_zero_day`. The original date-math classification is preserved separately for sensitivity analysis. The remaining same-day CVEs that could not be confirmed through any tier were classified as `post_nvd`. See Section 6 for discussion of these indeterminate cases. Subsequent enrichment passes and dataset updates have refined this count to 114 CVEs currently.

Limitations: Some advisory pages require JavaScript to load and could not be retrieved automatically. For CVEs assigned by MITRE or VulnCheck (as a CNA of last resort), no vendor advisory exists, so no language check is possible. When no advisory is found, the CVE is conservatively left as a non-zero-day. The true zero-day rate is therefore a lower bound.

Method 3: WordPress Discovery-Pattern Classification

> **MECHANISM**
>
> CVE is assigned by Wordfence or Patchstack, with exploit date, patch date, and NVD publication date all on the same calendar day.

Wordfence and Patchstack are WordPress security companies whose firewall products monitor millions of sites. When their monitoring detects active attacks against a WordPress plugin, they coordinate a patch with the plugin author, assign the CVE, publish their advisory, and notify CISA all on the same day. This means the patch date, CVE publication date, and catalog date always land on the same calendar day for every CVE they discover.

This creates a consistent pattern where exploitation predates the patch, but date comparison alone cannot detect it. The CVE didn't exist until the day exploitation was found and the patch was issued, so the dates are identical by construction. Method 3 recognizes this pattern as evidence of pre-patch exploitation. It applies only when all three dates match. WordPress CVEs where exploitation was recorded after the patch are unaffected.

This is a category-level classification based on how these organizations operate, rather than reviewing each CVE individually.

> **SENSITIVITY NOTE**
>
> Method 3 is the most methodologically distinctive of the three. If a conservative stance is preferred (only per-CVE advisory evidence accepted), it can be omitted. Doing so lowers the true zero-day rate by approximately 2-3 percentage points and leaves all other findings unchanged.

6. Indeterminate Cases

After applying all three methods and subsequent patch date enrichment passes, **114 CVEs with all three dates on the same day** remain classified as `post_nvd` because no vendor advisory could be confirmed. Most of these were assigned by MITRE or VulnCheck acting as fallback assigners for vendors without their own CVE programs. Neither organization publishes vendor advisories, so Method 2 cannot be applied and Method 3 does not apply (it is specific to Wordfence and Patchstack's documented process).

These 114 cases very likely include genuine zero-days, but there is no systematic way to confirm that without per-CVE advisory evidence. They are conservatively left as `post_nvd`, which means the 18.9% true zero-day rate is a lower bound.

7. Known Limitations and Caveats

Exploit signal dates are proxies, not ground truth. The exploit date is the earliest date on which exploitation was recorded in a public catalog, not the date exploitation began. Actual first-exploitation dates are generally unknown and often substantially earlier than catalog recording dates. The true zero-day rate is a lower bound on the actual rate.

Calendar-day resolution. All dates in this dataset are at calendar-day precision. The sequence of events within a calendar day cannot be determined from public data. Same-day cases require Methods 2 and 3. Those that cannot be resolved remain conservatively classified as `post_nvd`.

CISA KEV pre-2021 backfill. 31 CVEs in the dataset have an exploitation date of 2021-11-03 (the day CISA KEV launched), reflecting retroactive catalog additions without original exploitation date records. These should be excluded from any analysis that depends on precise pre-2022 exploitation timing (identifiable by `exploit_date = '2021-11-03'`).

Patch date reliability varies by source. MSRC dates (Patch Tuesday) and Apple HT advisory dates are high-confidence, tied to specific software release events. CVE API dates for vendor CNA CVEs are generally accurate. CVE API dates for MITRE-assigned CVEs carry more uncertainty because they reflect when the CVE record was created, which may not correspond to when a fix was available.

114 indeterminate cases. Triple-same-day CVEs assigned by MITRE or VulnCheck with no confirmable advisory page remain as `post_nvd`. If all 114 were reclassified as true zero-days, the true zero-day rate would rise by approximately 3 percentage points. The current figure is a defensible lower bound.

Recent years are incomplete in different ways for each document. For zero-day counts, the 2025 figure may still rise as CISA continues adding recently discovered cases, and 2026 (71 through July 15, 2026) covers only a partial year. For defender window medians, the 2024, 2025, and 2026 values are minimums rather than final figures. Exploitations that happened slowly and have not yet been cataloged will push those medians higher once recorded. The 2023 median (327 days) is the most reliable recent benchmark for the defender window.

Dataset selection bias. CISA KEV and VulnCheck KEV overrepresent severe, well-documented exploitation. The 18.9% true zero-day rate applies to "CVEs confirmed exploited and cataloged by CISA or VulnCheck," not "all CVEs exploited in the wild." The selection bias makes the 18.9% true zero-day rate conservative. Uncatalogued exploitation of less-tracked products is not represented.

WordPress distorts vendor and CWE comparisons. WordPress plugin CVEs carry a disproportionately high true zero-day rate that is a direct artifact of how Wordfence and Patchstack operate (see Method 3). CWE charts will show CWE-79 (XSS) and CWE-862 (Missing Authorization) as the top flaw types largely because those are the dominant flaw types

in WordPress plugin CVEs, not because they are the most exploited flaw types in traditional software.

8. Sensitivity Analysis

SCENARIO	APPROACH	EXPECTED EFFECT ON TRUE ZERO-DAY RATE
Without Method 3	Exclude Wordfence and Patchstack triple-same-day reclassifications; restore their original date-math classification	Drops ~2-3 percentage points (from 18.9% to ~16-17%)
Without CISA KEV backfill CVEs	Exclude the 31 CVEs with exploitation dates of 2021-11-03	Minimal; removes 31 CVEs with unreliable pre-2022 dates
Without WordPress CVEs	Exclude CVEs assigned by Wordfence, Patchstack, WPScan, and Automattic	True zero-day rate drops substantially; use for non-WP segment analysis
Full dataset vs. 2018-2026	Include all CVEs from 2010 onward	Full-dataset true zero-day rate = 17.4% (795 of 4,571 including pre-2018 CVEs)
Upper bound	Reclassify all 114 remaining indeterminate triple-same-day cases as true zero-days	Rises ~3 percentage points