runZero

DIVINING RISK

Deciphering Signals from Vulnerability Scores

CVSS EPSS SSVC

Executive Summary

Modern security teams rely on vulnerability scoring systems to prioritize patching and response efforts, but the three most popular scoring systems, CVSS, EPSS, and SSVC, were designed with different goals and assumptions. This paper examines these systems and offers a critical assessment of how they are used, misused, and misunderstood, while surfacing the practical value they do offer defenders.

CVSS (Common Vulnerability Scoring System) is a severity scoring system designed to describe the technical impact of a vulnerability in a vacuum. It is useful for measuring inherent characteristics, such as general attack vector, likely impacts, and preconditions required, but does not indicate real-world risk. Despite this, CVSS is the de facto prioritization and risk metric. As we will see below, it also creates a false sense of precision, while simultaneously overwhelming defenders with an impossibly large list of vulnerabilities that purport to require immediate attention.

EPSS (Exploit Prediction Scoring System) uses machine learning to estimate the likelihood that a vulnerability will be exploited in the next 30 days with the goal of providing the real-world risk context missing in CVSS. While not perfect, it offers a meaningful signal, especially when considered in a time series of scores. However, it's an opaque system, and its probabilistic nature can be easily misinterpreted by those unfamiliar with the underlying data science.

SSVC (Stakeholder-Specific Vulnerability Categorization) is a decision framework, and does not provide an objective "score." Instead, it helps analysts incorporate environmental context, such as mission criticality and asset exposure, into prioritization decisions. SSVC is a powerful tool, but it requires organizations to have deep visibility into their environments and asset inventory, which most teams lack.

CVSS

Common Vulnerability Scoring System

A multidimensional matrix of attributes about vulnerability features.

EPSS

Exploit Prediction Scoring System

A machine learning algorithm with thousands of inputs about vulnerabilities and exploitation.

SSVC

Stakeholder-Specific Vulnerability Categorization

A hyperlocal, five-step decision tree.

Key Findings

Studying the contemporary outputs for the three public, common vulnerability scoring systems in April 2025 revealed that all three systems fall short in providing an easy, unambiguous risk "score." However, each system offers some less-obvious insights into vulnerability criticality in its own way. For more specific technical details on the methodology used, please refer to Appendix A. In summary:

CVSS: Score distributions remain consistent over time, giving the illusion of stability while offering little in terms of predictive value or prioritization guidance. However, CVSS attack vectors can help analysts quickly discount vulnerabilities with a low likelihood of attack.

EPSS: This system offers strong signal value, particularly for CVEs that later jump up in score. These daily percentile shifts can act as an early warning, but require careful monitoring to be actionable.

SSVC: SSVC highlights a critical operational gap: most organizations are still struggling with basic asset management and exploit intelligence, which makes comprehensive environmental triage difficult in practice. Proof-of-concept (PoC) exploits remain a crucial, but underutilized, indicator of imminent threat. Projects like CISA's Vulnrichment are helping validate and expose this data to practitioners.

Ultimately, defenders must navigate a landscape of fractured signals and overloaded tooling. This paper advocates for a more grounded and nuanced approach to vulnerability triage — one that uses elements of CVSS, EPSS, and SSVC not as definitive answers, but as signals, hints, and even omens if you will. No scoring system offers binary and robotic "patch this, ignore that" directives. As defenders, we must combine these systems with our own experience, expertise, judgment, and continuous discovery of systems we're actually trying to protect.

A Triad of Vulnerability Scoring Systems

Modern technical vulnerability management relies on three open, common standards for gauging the seriousness of vulnerabilities:

- **CVSS** (Common Vulnerability Scoring System)
- **EPSS** (Exploit Prediction Scoring System)
- **SSVC** (Stakeholder-Specific Vulnerability Categorization)

Since their releases, all three systems have undergone significant revisions, and for the purposes of this paper, written in May of 2025, we will be concentrating on version 3.1 of CVSS (released in June of 2019), version 4 of EPSS (released in March of 2025), and version 2 of SSVC (released in July of 2023).

Note that while CVSS in particular has had a major version update to version 4.0 (released in November of 2023), this later version has not seen significant enough uptake in the vulnerability management industry to warrant focus. That said, this paper will briefly touch on some of the differences between version 3 and version 4 below.

CVSS: The Foundational and Familiar

As the first widely accepted vulnerability rating system available, CVSS is a cornerstone of traditional vulnerability management. Released in February of 2005, it is arguably the most popular scoring system, producing a number between 0 and 10, based on eight core attributes with some optional ones, common to all technical vulnerabilities:

1. **Attack Vector**
   (Network, Adjacent, Local, or Physical)
2. **Attack Complexity**
   (Low or High)
3. **Privileges Required**
   (None, Low, or High)
4. **User Interaction**
   (None or Required)
5. **Scope**
   (Unchanged or Changed)
6. **Confidentiality Impact**
   (None, Low, or High)
7. **Integrity Impact**
   (None, Low, or High)
8. **Availability Impact**
   (None, Low, or High)

There are many more possible attributes that a CVSS score can be derived from. For example, the "Temporal Score" domain addresses attributes which are likely to change over time, and covers questions like whether exploit code is available, patches are available, and confidence in the vulnerability reporter. The "Environmental Score" is intended for use in individual sites where the vulnerable component is found, offering modifications to the above base attributes.

Version 4 of CVSS builds on the foundations of Version 3, introducing new elements like Attack Requirements (which describe preconditions that would likely hamper exploitation attempts), and refining Version 3's "Scope" attribute by expressing it with a set of "Subsequent System Impact Metrics."

In practice, though, neither the extended attributes of Version 3 nor the additions in Version 4 are widely used or referenced when discussing a given vulnerability's base score. Instead, vulnerabilities tend to be published either with a base Version 3 assessment from the provider of the affected component, the original researcher and reporter of the vulnerability, or a more centralized clearinghouse of vulnerability intelligence such as the US National Vulnerability Database (NVD) or the US Cybersecurity and Infrastructure Security Agency Authorized Data Publisher (CISA-ADP, also known as the Vulnrichment project).

CVSS remains the most widely accepted system across risk management frameworks and traditional vulnerability management solutions. With only a little practice, most security professionals have no trouble eyeballing a vulnerability disclosure and feeling confident in stating if a particular vulnerability is exploitable over a network (thus, **AV:N**), or if a successful exploitation would result in the attacker seizing complete control over the affected system (thus, **I:H**).

FIRST (Forum of Incident Response and Security Teams) maintains convenient online calculators that generate final scores, designated as Critical, High, Medium, and Low, depending on where the number ends up.

Additionally, the full "Vector String" is a convenient shorthand to describe how a score is derived. In this example: "`CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N`," the vector string represents a 9.1, or Critical vulnerability, and describes the eight base attributes covering exploitable conditions, attack vectors, and impact of the vulnerability.

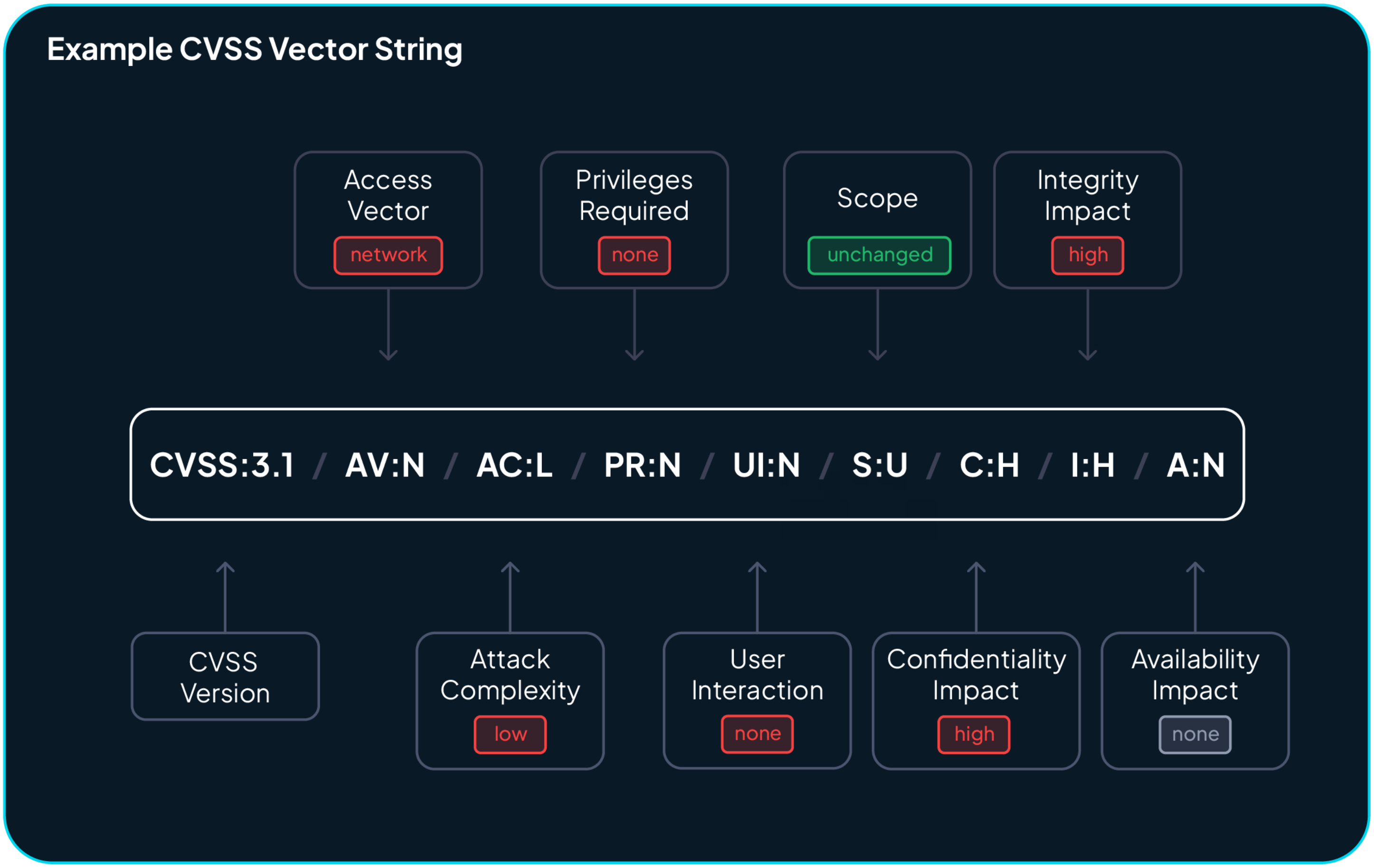


Figure 1: Detail on an example CVSS vector string

At its core, CVSS was built for consistency and objectivity to help standardize how we describe software vulnerabilities across teams, organizations, and organizational processes, enabling decision makers to make reasonable, well-informed recommendations based on the severity of a vulnerability. This consistency also helps normalize how people on different teams or at different organizations discuss a given vulnerability, regardless of their unique strategies for managing the day-to-day flow of vulnerability remediation.

However, all is not perfect in CVSS, which is why other systems for measuring the "threatiness" of vulnerabilities have emerged.

EPSS: The Statistically-Driven Newcomer

FIRST introduced EPSS in April 2021 to approach vulnerability risk from a different angle – not based on severity, but on the likelihood of exploitation in the wild.

Instead of analyzing a vulnerability and breaking it down by a set of defined attributes, EPSS predicts the likelihood — expressed as a probability, between 0 and 1 — that a given vulnerability will be exploited in the next 30 days. It does this by running a centralized machine learning model that measures thousands of signals about vulnerabilities, as well as measuring evidence of exploitation activity collected from a variety of sources. Some of these signals are public; others are private and come from proprietary data-sharing agreements.

Many attributes contribute to this vulnerability assessment model, such as:

- **CVSS vectors** are common inputs; many (though not all) Common Vulnerabilities and Exposures (CVE) records include them. If not, they tend to acquire CVSS attributes through a variety of methods from either NVD or CISA-ADP shortly after publication, either directly as part of the CVE record or in secondary sources such as vulnerability management systems.
- **CWE IDs**, or Common Weakness Enumeration Identifiers, are also somewhat common attributes of CVE records, but again, sometimes are not immediately or directly reflected in a particular CVE record.
- Mentions or or implementations of vulnerabilities in highly-regarded forums or tooling — such as the **CISA KEV** and **Google's Project Zero** tracker, and the open source **Metasploit** and **Nuclei** projects — also boost a vulnerability's profile within the model.

There are dozens of sources of vulnerability material, and most of those preferred by practitioners end up as data sources for EPSS.

When it comes to measuring exploitation activity, EPSS ingests alerts and traffic gathered from a variety of real-world sources in the wild, including:

- Honeypot hits
- IDS (Intrusion Detection System) alerts
- Endpoint detection and response agents
- Daily background noise from across the internet provided by cybersecurity vendors and early warning systems operated by the likes of **Fortinet, GreyNoise, Shadowserver**, and others.

EPSS also identifies and measures real-time social media, blogs, and news article chatter among information security practitioners, and uses these public sources to spot trends in exploitation activity associated with specific CVEs.

The exact makeup of what constitutes "vulnerability information" and "exploitation activity" is largely unknown, and indeed, is unknowable. This is the nature of machine learning algorithms. There is not one master case statement or if/then tree that determines the final or relative EPSS or percentile scores; instead, there is a stupendous amount of complicated algebra involved. Specifically, EPSS makes use of gradient boosted trees, where many, many weak little predictors add up and become surprisingly effective at spotting trends.

The outcome is a daily-updated prediction of exploitation probability for each CVE available for download via FIRST.org or Cyentia. While not perfect, EPSS has been shown to be valuable in gauging the exploitability of CVE-identified vulnerabilities to inform decision-making.

But what if math isn't your thing, or if you have trust issues with statistical approaches? Enter SSVC.

SSVC: The Human-Centric Decision Framework

SSVC, first introduced in November 2019, provides a decidedly non-mathematical decision tree framework for arriving at a predetermined set of outcomes of "what to do" about a given vulnerability.

SSVC is maintained and promulgated by CISA and the Software Engineering Institute (SEI) of Carnegie Mellon University, and takes a decidedly local approach to assessing vulnerabilities. SSVC's goals are less about standardizing discussions about vulnerabilities or predicting future exploitation, and instead focus on the potential impact of a newly discovered vulnerability *on your environment*.

To do this, SSVC poses a series of questions:

- Is the vulnerability being actively exploited, or is an exploit publicly available?
- Could that exploit be automated, even in theory?
- What's the technical impact if that exploit succeeds?

From there, SSVC turns to its most subjective — and organization-specific — elements: the "Mission and Well-Being" impact of a successful exploit. These depend entirely on the analyst's own environment.

For example, if the vulnerability is only present in the Windows operating system, but your environment consists entirely of Apple Macbooks running on Google Cloud, mission impact is pretty minimal. The well-being assessment in particular is often reserved for industrial control systems (ICS), environmental controls, weapons platforms – things that can damage the physical infrastructure or endanger lives if things go sideways.

Because of this local focus, SSVC doesn't aim to produce one universal score that fits every environment. But just like CVSS, analysts can usually reach agreement on more general elements like exploitation, automatability, and technical impact. In fact, many CVE records now include these values, thanks to CISA's Vulnrichment project, leaving only Mission and Well-Being impacts for analysts to assess locally.

Once general criteria are established, SSVC yields one of four decisions:

1. **Track:** no immediate action required; just keep an eye on it
2. **Monitor:** watch for changes that could raise the risk
3. **Attend**: investigate or mitigate
4. **Act:** drop everything and respond immediately

These decision points are flexible depending on who is doing the triage — but in general, "Track" means the issue is parked, and "Act" means it's time to call stakeholders, prioritize patching, and prepare crisis communications in the event of an imminent compromise.

SSVC is starting to grow in popularity. It's the primary risk assessment system used by CISA on behalf of the US government's federal civilian executive branch agencies and is seeing some uptake in private industry. There is a public calculator available on CISA's website that can be used to step through the decision process and illustrate the entire decision tree.

A Deeper Dive Into Dynamics and Efficacy

With all three systems defined, including the complementary goals they pursue, how do they fare at what they promise to deliver? Should IT organizations drop what they're doing and patch every CVSS-scored critical vulnerability? Does EPSS predict exploitation across the internet with the accuracy of the morning weather? Does SSVC provide IT teams with enough triage value to isolate their exposures before they get hit? Unfortunately, no.

While each system provides useful signals, human insight, experience, and expertise are required to make consistently good calls on which vulnerabilities warrant our immediate attention. CVSS, EPSS, and SSVC help, of course; the business of exposure management is not just gut instinct and prognostication. However, the rest of this paper will break down some of the mythology and misunderstandings around these scoring models, as well as make a case for some of the signals that can be teased out. Managing vulnerabilities today is less about cold calculation and more about cultivating a sixth sense — learning to read the faint, sometimes conflicting signals hidden inside our standard scoring systems.

CVSS: The Vulnerability Echo Chamber

CVSS's purported strength is in its consistency. However, when looked at in the aggregate, this consistency is one of its biggest weaknesses. Our experience as a software-developing civilization seems to imply that software vulnerabilities are rare and largely unpredictable when it comes to severity or impact. And yet, consider these three graphs below ↓. Can you spot the differences?

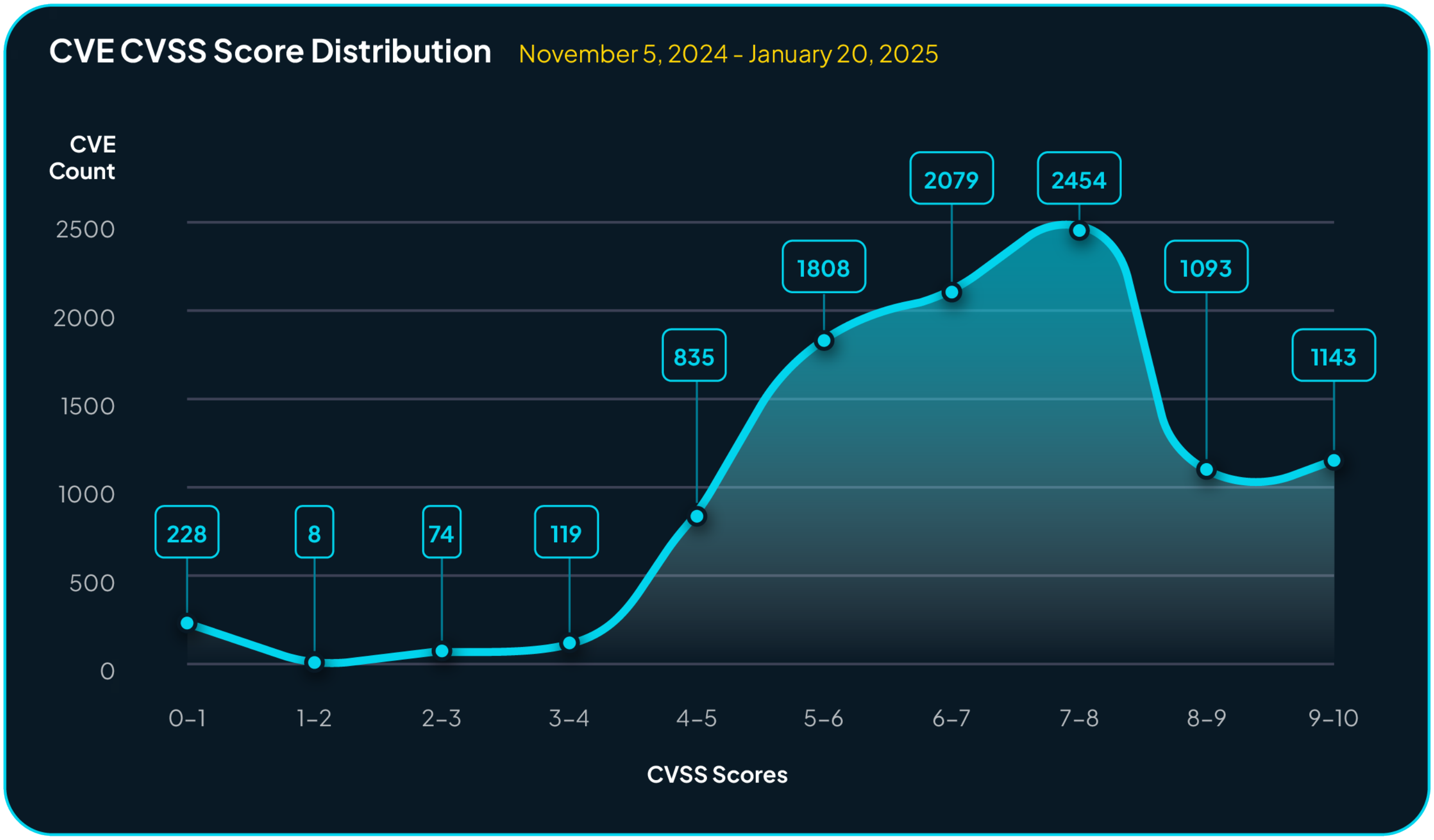


Figure 2: 2024–11–05 through 2025–01–20 (data source: cvedetails.com)

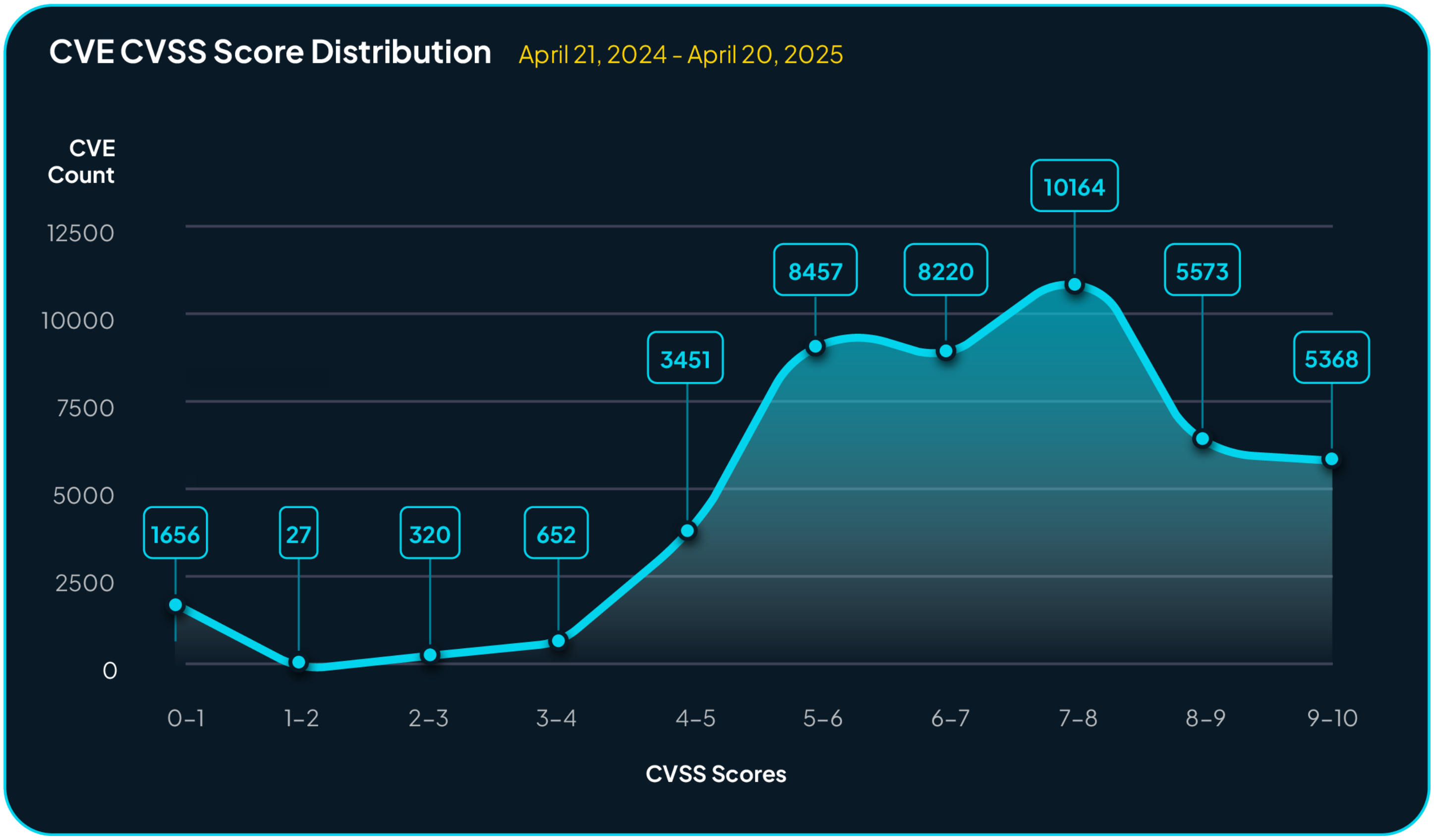


Figure 3: 2024–04–21 through 2025–04–20 (data source: cvedetails.com)

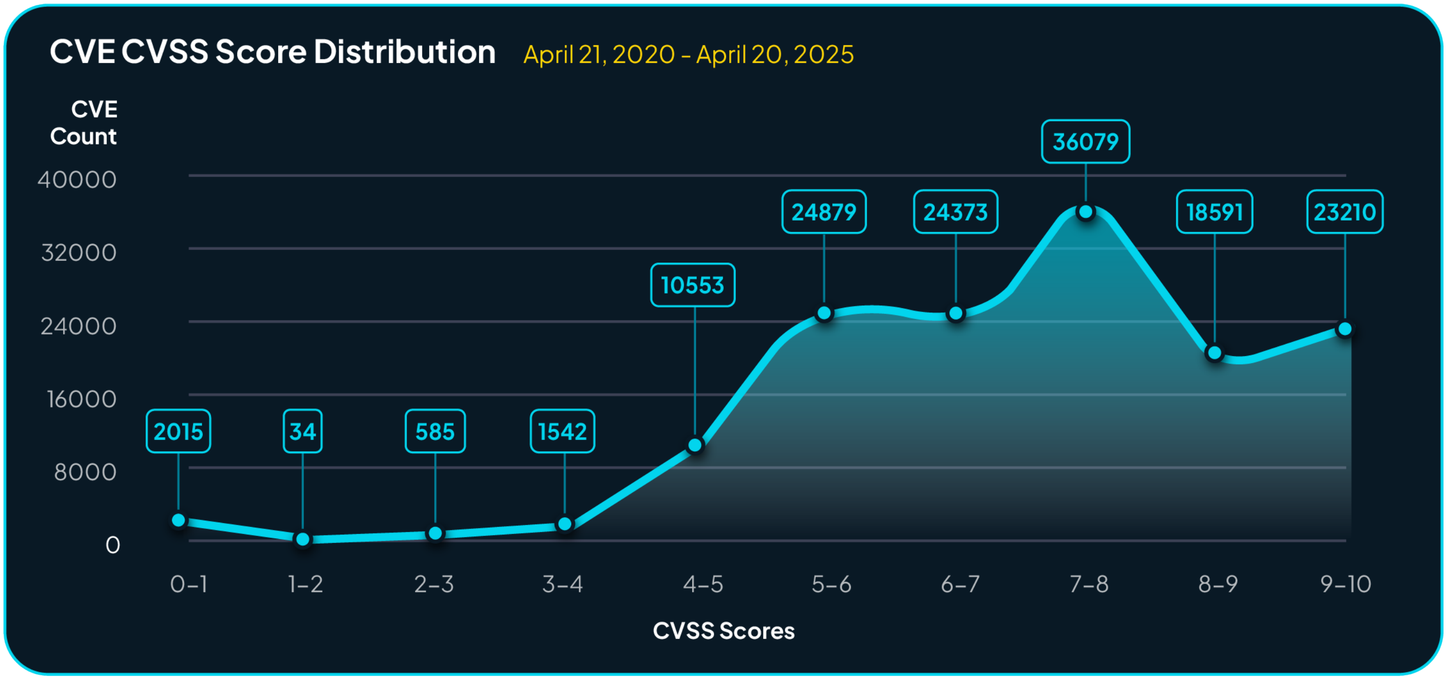


Figure 4: 2020–04–21 through 2025–04–20 (data source: cvedetails.com)

These charts illustrate three datasets: a recent, randomly chosen 76–day period, a twelve-month view from April 2024 to April 2025, and a full five-year span from 2020 through 2025. While there's a slight (but notable) trend toward better identification of extremely low and extremely high CVSS scores, the overall shape of the distribution remains stable, period to period. No matter the time slice, the peak sits squarely in the 7–to–8 range, with a quick drop and a plateau in the 8–to–10 band. Below are the same three datasets, expressed as a bar chart to illustrate the similarities of the distribution of CVSS scores across all three periods.

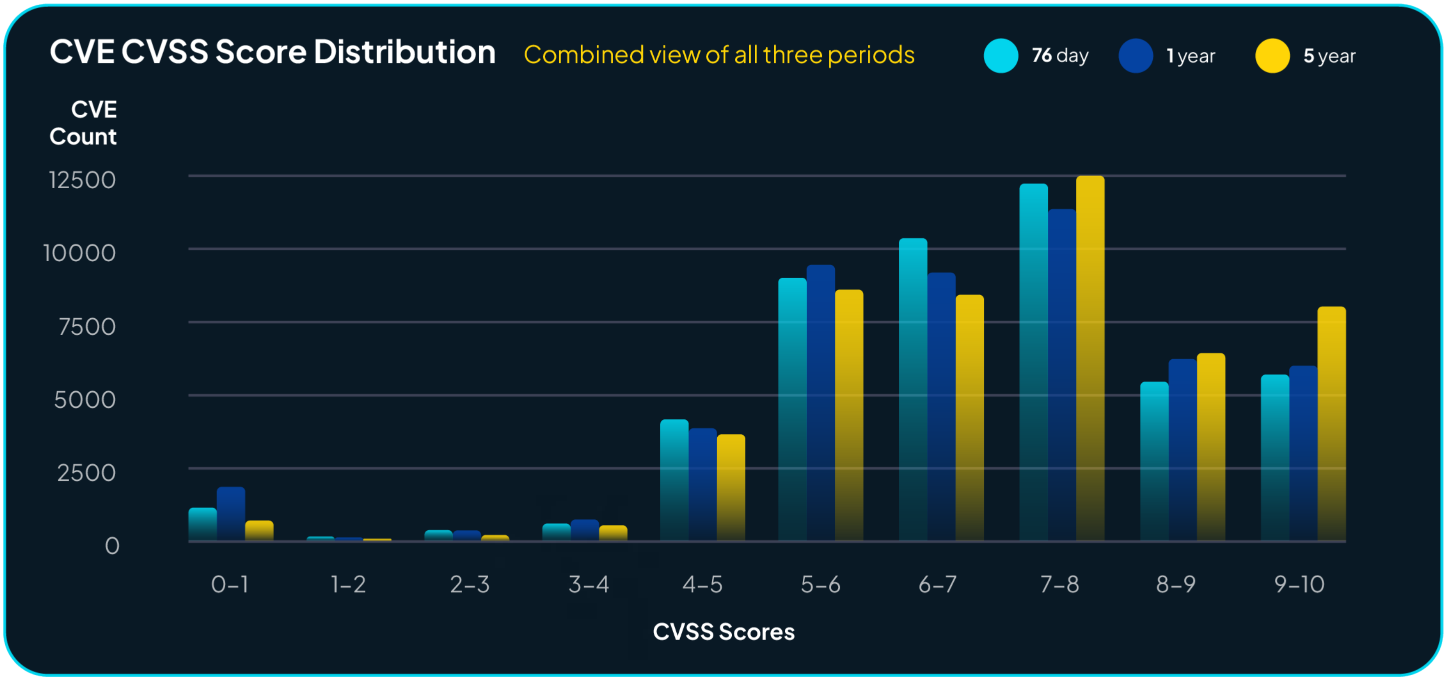


Figure 5: All three periods expressed in one bar chart (data source: cvedetails.com)

This fractal effect – zoom in, zoom out, same shape – suggests that CVSS isn't revealing the chaos of the vulnerability landscape. It's reflecting the structure of its own scoring rubric. In other words, what we're seeing over time isn't the unpredictable nature of vulnerabilities; it's CVSS's internal logic, playing out again and again.

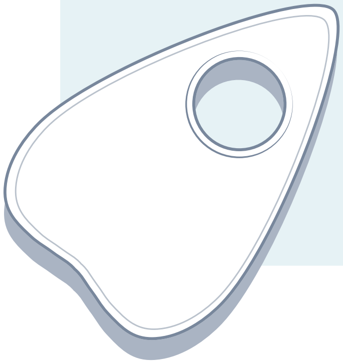
If we have an intuition that leads to a scoring system for vulnerabilities that claims a range from zero to 10, we should expect one of two outcomes:

1. A relatively smooth bell curve, with most vulnerabilities in the middle of the extremely low and high values
2. An unpredictable, uniformly distributed scattering of scores reflecting real-world unpredictability

Yet neither is the case. We see this pattern of mostly high severity vulnerabilities, with a sizable clump of criticals on the right.

This skew toward criticality may be easily explained. It's likely an artifact of the costs involved in identifying, verifying, writing, submitting, and publishing vulnerabilities. It may not be worth it for security researchers and software producers to go through this exercise for mere low and medium-severity vulnerabilities. By themselves, these issues aren't worth publicizing — or possibly even fixing — and so they go unreported.

This leads to a conclusion that is possibly the scariest revelation this paper has to offer:



There almost certainly exists a vast trove of vulnerabilities going unreported, undetected, and unfixed, and it's expanding with time as new software is released and old software goes end-of-life.

This fractal skew was described quite ably in Jacques Chester's paper, "A Closer Look at CVSS Scores" (June of 2022) and earlier alluded to in Henry Howland's paper, "CVSS: Ubiquitous and Broken" (February 2022). After almost three years since those papers were published, this effect remains. Despite advances in secure software development, more common use of design and development frameworks, and better transparency and participation in the CVE Program, we are still reporting about the same proportions of high to critically-rated software vulnerabilities. CVSS is still ubiquitous, but it's not the only thing that's broken. Everything is.

The saving grace of CVSS is the attack vector string; this is where seasoned security professionals look first to gut-check the severity of a vulnerability in a vacuum. Or rather, where you can look to quickly discount vulnerabilities.

For example:

- If the attack vector is physical (AV:P), and assuming you believe the CVSS data, your attacker needs to have hands-on-keyboard.
- If it's local (AV:L), there's some limited remote accessibility, but you're still not looking at a straight-shot attack from the internet. Defense in depth practices often reduce these vulnerabilities to non-issues entirely.
- Similarly, a privileges-required rating of high (PR:H) tells you the attacker needs to not only be authenticated, but also to be authenticated with higher privileges, such as "application admin" level privileges. These kinds of vulnerabilities are good for further privilege escalation attacks, but they don't in and of themselves provide much in the way of initial access.

This is not to say that IT security folks should go to sleep on these vulnerabilities; after all, someone did go to the trouble of reporting, and hopefully, fixing them — which signals they aren't meaningless. But if you're looking to use CVSS in a useful way that goes beyond compliance, skip severity scores entirely and focus on the vectors.

EPSS: High-Value Movers and Shakers

EPSS takes a fundamentally different approach from CVSS. Rather than scoring severity, it produces statistical, probabilistic values that IT defenders can use to determine the chance that a given vulnerability "should" be exploited in the wild.

As explained above, EPSS compares many attributes of known vulnerabilities listed in the CVE corpus — ranging from CWE identifiers, CVSS scores, description keywords, number and quality of references, and so on. Then, it looks for patterns among the many signals it uses to determine exploitation activity. However, taking this at face value, EPSS can quickly lead to some nonsensical results.

For example, by taking the entirety of EPSS scores dated March 17, 2025 (the first day that EPSSv4 scores were available), and running them through a standard Monte Carlo simulation of several thousand trials, we "should" expect about 10,000 distinct CVEs to be exploited. Here's the summary set of all EPSS scores (as of March 17, 2025): →

unique-deciles	decile-counts
0–10%	251,242
10–20%	6852
20–30%	3381
30–40%	2019
40–50%	1482
50–60%	1385
60–70%	1368
70–80%	1328
80–90%	1370
90–100%	1180

Table 1: EPSS broken down by deciles

Across the entire set of over 270,000 vulnerabilities scored, the distribution looked like this:

- About 250,000 CVEs had an EPSS probability less than 10%,
- About a thousand CVEs had a probability over 90%

While we don't know precisely which ones will be exploited, or where, or what exploitation even looks like — we should expect, in theory, over 30 days that:

- The CVEs rated at around 1% should be exploited about one in a hundred times
- The CVEs rated 90% should be exploited nine times out of ten

Here's what the Monte Carlo simulations have to say:

```
[*] Average number of exploited vulnerabilities over 10000 simulations: 10094
[*] Standard deviation: 127.27
[*] Empirical Rule for standard deviations:
    68% of simulations fall between 9967 and 10221
    95% of simulations fall between 9839 and 10349
    99.7% of simulations fall between 9712 and 10476
```

And here we are, in the future, and we can see that the internet did not halt and catch fire from the heat of all this predicted exploitation. (A lot of other bad things happened on and off the internet during this period, but total infrastructure meltdown wasn't one of them.)

Ten thousand distinct CVEs leading to popped shells and leaked credentials over thirty days would be a truly stunning figure. Instead, we saw over the same period (March 17 through April 15), a total of **21 vulnerabilities** added to the CISA KEV. Not hundreds, not thousands, not even 50. Even if we (rightly) assume that KEV does not have total visibility into all exploitation the world over, and not all exploits rate for the KEV for various reasons, it's still off by a staggering two orders of magnitude, or missing about 99.8% of all exploitation events (not counting multiple exploits hitting the same CVE, which of course happens routinely). In the entirety of CISA's KEV history, only about 1300 CVEs have been added.

The reason for this disparity between predicted exploitation and the observed reality is that EPSS does not actually predict exploitation in the normal sense of the term. Instead, it predicts exploitation activity, as explained in the most recent paper, "Enhancing Vulnerability Prioritization: Data-Driven Exploit Predictions with Community-Driven Insights" (February, 2023) by Jay Jacobs et al. In short, EPSS counts a multitude of signals as exploitation activity, going beyond popped shells to include things like:

- IDS alerts
- Honeypot hits
- Researcher chatter that mentions particular CVEs

This distinction is subtle and often overlooked when considering the accuracy of EPSS's predictions. (In fairness, the CISA KEV is often criticized for its lack of exploitation attribution for precisely the opposite reason; the bar for exploitation is intentionally quite high and depends a lot on the veracity of the reporter, the presence of a victim, and other believable attacker activity.)

The EPSS FAQ states plainly that "EPSS is best used when there is no other evidence of active exploitation." In other words, if you watch the threat actor exfiltrate your database by using a recently disclosed vulnerability, trust your observation over EPSS scores. How does that advice hold up? The table below lists the top twenty EPSS-rated CVEs on March 17, 2025, as a spot check on EPSS's on-its-face veracity.

CVE	EPSS
CVE-2024–27198	0.94582
CVE-2023–42793	0.94575
CVE-2023–35078	0.94496
CVE-2024–27199	0.94496
CVE-2018–7600	0.94489
CVE-2019–3396	0.94486
CVE-2021–22986	0.94485
CVE-2021–44228	0.94482
CVE-2021–22205	0.94479
CVE-2014–0160	0.94477
CVE-2019–0708	0.94475
CVE-2022–22947	0.94474
CVE-2022–22963	0.94474
CVE-2017–8917	0.94471
CVE-2018–13379	0.94471
CVE-2018–1000861	0.94469
CVE-2020–1938	0.94469
CVE-2022–46169	0.94469
CVE-2021–22005	0.94467
CVE-2019–15107	0.94461

Table 2: Top 20 EPSS-scored CVEs

As it turns out, all issues but one, CVE-2017–8917 (a SQL injection in Joomla! before 3.7.1) are listed on either CISA KEV or the VulnCheck KEV. One other, CVE-2024–27199, a path traversal issue in JetBrains TeamCity, appears only on the VulnCheck KEV, and not the CISA KEV. Furthermore, all of these issues landed on the two KEVs well before March 17, 2025, as they were disclosed in many different years. While past exploitation does not equal a 100% chance, this vulnerability will be exploited in the next 30 days, it does seem to be a pretty good sign that something will very likely trip a detector related to these vulnerabilities.

So, how can we use EPSS effectively to make better practical predictions? When examining various aspects of EPSS, an interesting phenomenon emerges when comparing the day-to-day changes of an EPSS score.

CVE	Change	Date
CVE-2023–40477	0.81136	2025–04–15
CVE-2024–30568	0.8102	2025–03–28
CVE-2023–36255	0.80409	2025–03–26
CVE-2024–7314	0.75756	2025–04–05
CVE-2023–36824	0.74793	2025–04–15
CVE-2017–12637	0.72812	2025–03–20
CVE-2023–41249	0.69238	2025–04–15
CVE-2024–3080	0.68676	2025–03–22
CVE-2023–46042	0.65923	2025–04–15
CVE-2023–33735	0.65704	2025–04–15
CVE-2023–4568	0.65515	2025–04–15
CVE-2023–33404	0.65266	2025–04–15
CVE-2023–6016	0.65037	2025–04–15
CVE-2023–44443	0.64839	2025–04–08
CVE-2023–41064	0.64661	2025–04–15
CVE-2024–2387	0.64508	2025–03–25
CVE-2023–39108	0.64436	2025–04–15
CVE-2023–39110	0.64436	2025–04–15
CVE-2023–39109	0.64436	2025–04–15
CVE-2007–1277	0.63002	2025–03–30

Table 3: Top 20 positive movers from March 17 to April 15

This is a much more interesting list, and may well be closer to EPSS's goals of asserting imminent or very recent exploitation. It certainly indicates **new and unusual** exploitation activity worth paying attention to. While this effect was hinted at in Riana Parla's November 2024 paper, "Efficacy of EPSS in High Severity CVEs found in KEV," all but two of these CVEs (CVE-2017–12637 and CVE-2023–41074) are **not** listed on the CISA (or VulnCheck) KEV, despite having seen recent 50–point jumps in EPSS scores.

In fact, of all EPSS scores analyzed (about a quarter million CVEs), only 73 saw 50–point day-over-day jumps in EPSS scores over the course of the new EPSSv4 data sets from March 17 through April 15, 2025. Only one jumps out as unusually old, from 2007 (and describes a fairly unique backdoor case). The top mover of the period is CVE-2023–40477, a bug in WinRAR disclosed by ZDI back in 2023.

Here's what that vulnerability's sequence looks like in the time series:

```
CVE-2023-40477,0.12458,0.12458,0.26827,0.12458,0.12458,0.12458,0.12458,0.26827,0.12458,
0.12458,0.31593,0.12458,0.26827,0.12458,0.12458,0.12458,0.12458,0.12458,0.12458,0.12458,
0.12458,0.12458,0.12458,0.12458,0.12458,0.12458,0.12458,0.12458,0.12458,0.93594
```

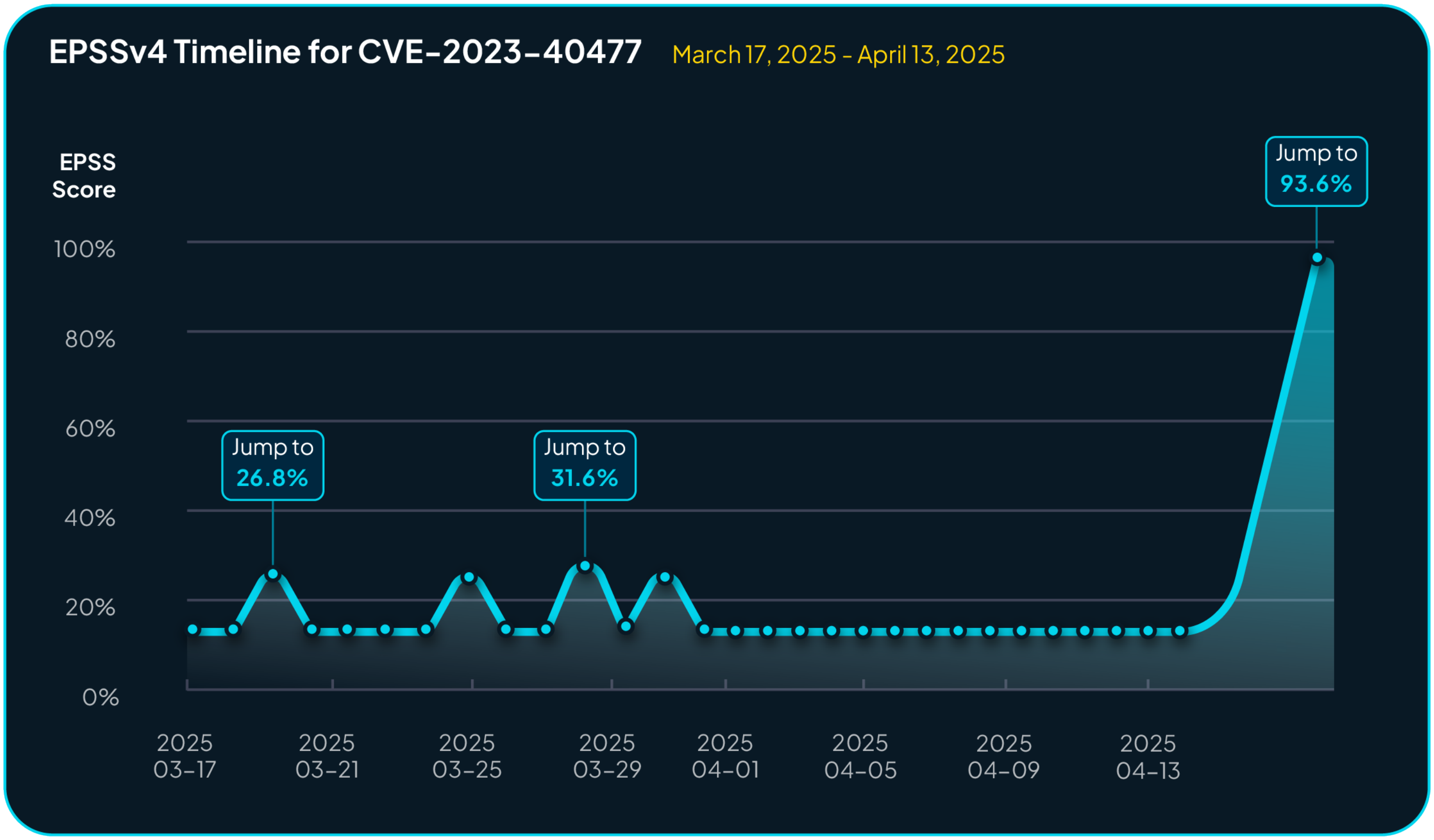


Figure 6: EPSSv4 scores for CVE-2023-40477 from March 17 through April 15, 2025

In this case, we can see that CVE-2023-40477 started its EPSSv4 life at a 12.5% chance of exploitation activity. Over the next 29 days, it jumped by 14 points on March 19, settled back, then rocketed up to 93.6% on April 15.

In terms of investigating interesting CVEs for action, the top 20 movers list appears to be a solid starting point. This CVE would be of particular interest, especially given that this kind of local-only vulnerability is very unlikely to appear in on-the-wire network logging.

More research should be done in this area. What's a useful threshold for a day-to-day change? Do more interesting CVEs shake out when looking at three-day or ten-day windows? What are we to make of sudden, dramatic drops in EPSS scores (of which there are over 300 in the same period). As EPSSv4 matures and the EPSS special interest group tinkers with the inputs, it's likely that large changes in score should smooth out, so watch for an update in about six months.

SSVC: Powerful in the Right Hands

As discussed, SSVC is a decision framework that gives analysts tools to categorize vulnerabilities based largely on environmental factors. While this approach is genuinely useful, it assumes you have a strong understanding of the mission-criticality and prevalence of affected assets. This is no small task, even for well-staffed enterprises. Using SSVC effectively, either alone or alongside other scoring systems, requires a level of environmental awareness that many teams simply don't have.

Close on the heels of asset management is a requirement for excellent exploit intelligence. Today, there are several well-known sources for proof-of-concept exploits, ranging from the venerable Metasploit Framework, to ProjectDiscovery's Nuclei and the many POCs available as one-shot scripts on GitHub. However, some discretion in inspecting random GitHub repositories is required.

A word of caution: the age of LLM-generated exploits is upon us, and it's becoming increasingly common to run across completely hallucinated exploits that either cannot possibly work in the way the vulnerability is described, or are so specific to a custom-configured target that it's useless in real-world exploitation activities.

The Vulnrichment project from CISA does help validate the existence of exploits. As of this writing, there are 1,871 CVEs dated from 2025 that are marked as "exploit: poc," which means that CISA itself has identified a working exploit available for those CVEs. While CISA isn't infallible (and accepts GitHub PRs for fixes when they get it wrong), the presence of a proof-of-concept exploit in this context means it was seen during SSVC triage. That alone makes these CVEs worth a second look.

While almost two thousand CVEs sounds like a lot to chase, it's a sharp cut from the over 15,000 CVEs published in 2025 at the time of writing. Of course, one can narrow the list further by filtering for both "Exploitation: PoC" **and** "Technical Impact: Total," which brings the list to 422 vulnerabilities at the time of this writing. But don't sleep on the ~1,200 that have known PoCs floating around.

If you are currently drowning in a flood of alerts, take heed. Vulnerability prioritization doesn't mean throwing out enough alerts to make the problem tractable. Solutions like runZero provide deep visibility into every asset across internal and external attack surfaces, enabling you to effectively detect and prioritize exposures based on real-world impact and context. This allows you to focus remediation activities on what's most likely to be exploited in your unique environment, which is the ultimate goal of all prioritization methodologies.

Finally, while you might expect "Exploitation: Active" to be an even better signal, that's really just the CISA KEV. You don't need to crawl through JSON-formatted vulnerability descriptions for that. Just head over to https://cisa.gov/kev and start fixing those. This policy of equivalence is likely to evolve as CISA tags active exploitation that doesn't make the KEV cut, but right now, it's a one-to-one relationship.

Conclusions

Vulnerability severity and risk scoring are still very much a dark art. And while good artists invent, great artists steal. Security teams are well-positioned to throw off the shackles of singular, context-free, point-in-time risk ratings and instead, use the tooling we have available to enhance our earned experience.

The most charitable view of vulnerability scoring as it stands today likens it to economic forecasting based on models like the Efficient Market Hypothesis or the Rational Actor Model: often useful, but built on flawed and oversimplified assumptions. Indeed, systems like CVSS, EPSS, and SSVC consistently fail to account for key human factors such as threat actor motivation, intent, opportunity, and skill. Crucially, while economic tools are grounded in decades of research, refinement, and empirical backtesting, vulnerability risk scoring is still a young and largely experimental discipline that requires the human touch to practice effectively.

Of course, the less generous perspective is that vulnerability scoring today resembles pseudosciences like phrenology, reflexology, or polygraphy. These junk systems claim to predict outcomes by measuring correlated but fundamentally unrelated attributes. In this view, scoring methods try to anticipate whether a vulnerability will be exploited by focusing on surface-level characteristics, while missing the deeper factors that truly drive attacker behavior.

This paper does not go this far. There are useful aspects of all three of the vulnerability scoring and categorization systems studied. In the hands of experts, they help us make better determinations about which vulnerabilities — or vulnerability chains — are likely to be truly catastrophic. But if we're to stay ahead of attackers, more study from more diverse perspectives in this area is required.

Appendix

The material created and collected for this paper has been made available on GitHub in the runZeroInc/divining-risk repository under a standard 2-clause BSD License.

Scripts provided there include:

- `build-epss-matrix.py`
  Builds a time series matrix of EPSS scores from a series of downloaded EPSS CSV files.
- `check-kev.sh`
  Checks a list of CVEs (nominally, the top 20 biggest movers in an EPSS time series) for KEV inclusion.
- `collect-epss-scores.sh`
  Downloads a series of EPSS CSV data files, based on a date range.
- `detect-significant-changes.py`
  Analyzes a Parquet-formatted set of EPSS scores and looks for those that move "a lot" over a number of days. By default, "a lot" is 50 points, and the period is one day.
- `find-cisa-ssvc-pocs.sh`
  Inspect a local clone of the cvelistV5 set, looking for those marked with proof-of-concept exploits available.
- `monte-carlo-epss.py`
  Takes a list of EPSS scores, and figures out how many scored vulnerabilities "should" be "exploited" in the next 30 days.

Data files include the set of EPSS scores collected from March 17, 2025, through April 15, 2025, a snapshot of the CISA and VulnCheck KEV lists, an XLSX export of specifically March 17th's EPSS data, and a set of significant changes and top-20 movers in EPSS scores over time.

Finally, specific technical instructions for gathering and parsing CVE, KEV, EPSS, and SSVC are provided in this repository. CVSS-based data was collected entirely from CVEDetails.com, and cited directly in the paper.

Acknowledgements

This paper would not have been possible without the invaluable conversations had both online and off with Jay Jacobs, Madison Oliver, Patrick Garrity, Art Manion, Robert "RSnake" Hansen, Jonathan "Jono" Spring, and Jon Cran. Thank you all so much for your time and expertise in this area.

RUNZERO
HOUR
Special Guest & EPSS Expert
ROB KING
Director of Security Research, runZero
JAY JACOBS
Founder at Empirical Security, Chief Data Scientist Emeritus, Founder at Cyentia Institute
TOD BEARDSLEY
VP of Security Research, runZero

Unpacking Vulnerability Scores

Join a spicy debate on findings from this report with vulnerability expert Jay Jacobs.

Watch the Episode

Great research sparks smarter defense

Stay ahead of the curve in exposure management with cutting-edge insights from runZero Research.

Explore runZero Research

See every asset. Know every risk.

No credentials, agents, or appliances required.

Try runZero Free

runZero provides a single source of truth for exposure management across your total attack surface: internal, external, IT, OT, IoT, mobile, and cloud.

Providing the most complete and accurate visibility into every asset and exposure, runZero helps you mitigate risks faster, meet compliance requirements, and ensure you continuously discover the assets and exposures that others miss.

250521