# Prediction Meets Patch Queues: Empirical Limits of EPSS-Only Prioritization Using CISA KEV Additions in 2025

Sergey Gordeychik
scadastrangelove@gmail.com

December 2025

**Abstract**

The Exploit Prediction Scoring System (EPSS) is often operationalized as a "patch gate": remediate vulnerabilities whose predicted exploitation probability exceeds a chosen threshold. The CISA Known Exploited Vulnerabilities (KEV) catalog represents a stricter notion of ground truth: exploitation has been observed and confirmed. Using all KEV entries added in 2025, end-of-year EPSS scores for the 2025 CVE population, and weekly EPSS snapshots throughout 2025, we quantify: (i) the *coverage gap* where EPSS scores are unavailable at the KEV decision time; (ii) the recall of exploited vulnerabilities under common EPSS thresholds; (iii) score drift and "catch-up" latency as EPSS evolves over time; and (iv) the operational consequences of EPSS-only intake under finite patch capacity. We further decompose the delay at the KEV inclusion moment into *signal lag* (EPSS not yet actionable) and *ops lag* (actionable but queued), showing that increasing patch capacity cannot compensate for a missing or late signal. Code, datasets: https://github.com/scadastrangelove/kev_vs_epss.

## 1 Introduction

Vulnerability management is frequently framed as an optimization problem with a single missing ingredient: a better score. Severity (CVSS) measures potential impact, not exploitation. EPSS attempts to fill the gap by estimating the probability of exploitation in the next 30 days. In practice, that probability is often treated as a control knob that converts a long tail of CVEs into a shorter patch list: "patch everything with EPSS $\geq \tau$".

Unfortunately, the real world is not a clean thresholding problem. First, exploitation is not uniformly observable; the KEV catalog itself is incomplete, but it is among the clearest public signals of exploitation in the wild. Second, patching is a service system with finite capacity and nontrivial lead times. In short: *prediction does not patch itself*. EPSS may be accurate in aggregate, yet still fail as an operational gate when (a) the score is missing at decision time, (b) the score wakes up late, or (c) the queue saturates.

This paper is a deliberately pragmatic investigation. We focus on the year 2025, treating CVEs added to CISA KEV in 2025 as "critical defects" a defender would prefer not to miss. We ask: if a defender relies on EPSS as the primary signal, what fraction of confirmed exploited vulnerabilities would have been caught at the time they mattered, and what portion of the delay is attributable to the signal versus operations?

# 2 Background

## 2.1 EPSS

EPSS produces a probability score in $[0, 1]$ intended to represent the likelihood of exploitation in the next 30 days. It also reports a percentile rank among all scored CVEs on that day. EPSS is described in Jacobs et al. and by the FIRST EPSS SIG documentation.

Two numbers are reported because they answer different operational questions:

- **Probability (EPSS):** an absolute estimate of near-term exploitation likelihood.
- **Percentile:** a relative rank—useful when the base rate of exploitation is low and one needs a sorting key rather than a single decisive cutoff.

## 2.2 KEV

CISA's KEV catalog is a curated list of vulnerabilities with evidence of exploitation in the wild. For reproducibility we use the public mirror maintained by CISA on GitHub.

## 2.3 Why "just use a metric" tends to disappoint

Metrics are attractive because they compress complexity into a single sortable number. That compression is operationally convenient, but it also hides heterogeneity (reachability, asset criticality) and combinatorial interactions across components. We return to this "instrument panel" problem in the Discussion, where we interpret our results through reliability, queueing, and communication-theory lenses.

# 3 Data and Methods

## 3.1 Datasets

We use three primary inputs:

1. **KEV 2025 additions:** all vulnerabilities with a `dateAdded` in 2025 extracted from the KEV dataset.
2. **EPSS daily and weekly snapshots:** EPSS scores retrieved via the FIRST API and the public daily snapshots hosted by Empirical Security (`epss_scores-YYYY-mm-dd.csv.gz`).[1]
3. **CVE-2025 universe:** all CVE IDs with year 2025 and their EPSS scores in the end-of-year snapshot (2025-12-29), used to estimate workload inflation under different thresholds.

## 3.2 Decision time and the "t=0" problem

For each KEV entry, we define decision time as the KEV `dateAdded`. This is intentionally conservative: by the time a vulnerability appears in KEV, exploitation has already been observed somewhere. If a score-based gate cannot catch items by that date, it is already behind.

[1] Snapshot hosting: https://epss.empiricalsecurity.com/

### 3.3 Threshold policies

We evaluate thresholds $\tau \in \{0.001, 0.01, 0.1\}$. A vulnerability is eligible if:

$$\mathbf{1}\{\text{EPSS}(v, t) \geq \tau\},$$

where $\mathbf{1}\{\cdot\}$ denotes the indicator function. Missing EPSS values are treated as not eligible.

### 3.4 Queueing model (patch capacity)

To connect prediction to operations, we model patching as a single-server queue with weekly time steps.

- **Arrivals:** newly-eligible vulnerabilities (first week in which EPSS crosses $\tau$).
- **Service:** a fixed weekly handling-time capacity (40h, 120h, 400h per week) converted into a number of items under a constant per-item handling time. We interpret this handling time as security-team effort (triage, coordination, change control, verification), not raw patch installation time. For backlog normalization we use 1 hour per vulnerability; this parameter is not intended as a universal estimate of patch time, but as a unit conversion that makes workload comparisons across thresholds interpretable, and backlog scales linearly with this value.
- **Scheduling:** first-come-first-served among eligible items (a deliberately naive "EPSS-only" process).

The key output is the fraction of KEV items completed as a function of time relative to `dateAdded`.

### 3.5 Lag decomposition

For each KEV item, we decompose delay into two parts:

$$\textbf{Signal lag} = t_{\text{eligible}} - t_{\text{KEV}}, \qquad \textbf{Ops lag} = t_{\text{patched}} - t_{\text{eligible}}.$$

At $t = 0$ (KEV inclusion), a miss can happen because the signal is not yet eligible (signal lag $> 0$) or because the item is eligible but stuck in the queue (ops lag $> 0$). This separation is operationally useful: one part is a sensor/model problem; the other is an operations/capacity problem.

## 4 Results

### 4.1 EPSS availability at decision time

Figure 1 quantifies the *coverage gap*: for a nontrivial subset of KEV additions, EPSS scores are unavailable at the KEV decision time. This is not a model-quality question; it is a data availability constraint that directly reduces the attainable recall of any EPSS-only gate. EPSS score availability differs slightly between the inclusion-date snapshot and the end-of-year snapshot. This reflects EPSS dataset updates/coverage drift rather than a change in our cohort.

### 4.2 Recall of KEV under EPSS thresholds

Figure 2 reports recall among KEV items at decision time for thresholds $\tau \in \{0.001, 0.01, 0.1\}$. Thresholds that are popular because they control workload also discard a large fraction of *confirmed exploited* vulnerabilities at the moment they become publicly confirmed.

### 4.3 Distributions and score drift inside KEV

Score distributions matter because operational decisions are often rank-based rather than threshold-based. Figure 3 shows that within KEV, the ECDF shifts substantially from KEV inclusion to an end-of-year snapshot—a reminder that the score is time-varying and may "catch up" after exploitation is already underway.

Figure 4 compares EPSS distributions for KEV-at-decision-time vs non-KEV CVEs in the same-year snapshot. KEV tends to be shifted to higher EPSS, but the overlap is substantial; this overlap is where gate policies become painful.

### 4.4 Catch-up latency

Figure 5 presents a Kaplan–Meier view of EPSS *growth* after KEV inclusion. We define a practical baseline $e_0$ as the first weekly EPSS snapshot on/after dateAdded, and measure the time until EPSS reaches a multiplicative target ($e(t) \geq g \cdot e_0$, with $g \in \{10, 100\}$). Items that never reach the target within the observation window are right-censored at the last snapshot.

### 4.5 A toy MTTR: "patch at end of week when taken into work"

A second-order effect of threshold policies is not only whether you patch, but *when*. Figures 6 and 7 summarize a simplified MTTR model where remediation completes at the end of the week in which the item is taken into work. Note the familiar trap: medians can look "fine" while tails become operationally ugly.

### 4.6 When the team is real: full queue, nine strategies

If EPSS is used as the only intake mechanism, the patch queue includes *all* CVEs above threshold, not only KEV. Figures 8 and 9 plot the fraction of KEV completed as a function of time relative to KEV inclusion, under nine combinations of thresholds and capacities. A central observation is that even with high capacity, the completion fraction at $t = 0$ can remain low—because many items were simply not eligible yet (*signal lag*).

### 4.7 What kills "t=0"? signal vs operations

Figure 10 answers the uncomfortable question: for a given policy, what fraction of KEV items are not completed by $t = 0$ because EPSS is not yet actionable, and what fraction is "actionable but queued"? This is where the discussion stops being religious and becomes actionable: you can buy capacity to fight ops lag; you cannot buy your way out of a missing or late signal without changing the sensing/triage process. Figure 9 curves are computed over the subset that becomes eligible under the policy (hence denominators 243/236/192), while Figure 10 reports shares over all 245 KEV additions.

## 5 Discussion: reliability, queues and communications

### 5.1 Reliability framing: how many critical defects do we ship?

Treat each KEV addition as a "critical defect" discovered only after it has already escaped into the wild. A threshold policy induces a simple reliability question: what fraction of these defects remain unmitigated at decision time? Figure 10 shows that, for realistic capacity (40 items/week), the answer is dominated by signal rather than effort. At $\tau = 0.001$, nearly half of KEV additions

(48.6%) are invisible to the policy at $t = 0$ because EPSS has not yet crossed the threshold. This is an upper bound on what any queue can fix: capacity can reduce waiting, but it cannot remediate items that never enter the queue.

### 5.2 Queueing framing: how many "rich clients" leave?

Once a policy defines eligibility, it also defines the arrival process into the remediation queue. Low thresholds ($\tau = 0.001$) admit almost everything, which can be rational in a world with abundant automation—or catastrophic in the more common world where patching is a scarce, disruptive operation. In our stylized FCFS queue, $\tau = 0.001$ with capacity 40/week yields an ops-lag median of 97 days and a 90th percentile of 149 days for eligible items. Higher thresholds reduce arrivals and shrink ops lag (median $\approx 6$ days for $\tau \geq 0.01$), but they do so by shifting the loss from "waiting" to "never arriving". This is the cynical trade: you can keep the queue small by rejecting customers at the door.

### 5.3 Signal lag as a communication problem

EPSS behaves like a detector output transmitted over a noisy channel with delay. A threshold $\tau$ is a decision rule; signal lag is detection latency; missing EPSS values are dropouts. The practical implication is that "better scheduling" does not improve the detection layer, and "better detection" does not increase patch throughput. If we borrow the language of communications, the base-rate of exploitation is low, the feature channel is noisy, and the resulting signal-to-noise ratio is not high enough to support sharp early decisions at high thresholds. Empirically, in the 2025 KEV cohort the time until EPSS crosses the threshold has a p90 of 7 days for $\tau = 0.001$, but a median of 20 days for $\tau = 0.1$—which is simply too slow if the goal is to be correct by the week exploitation is confirmed.

### 5.4 Positioning relative to prior critiques and scoring-system conflicts

Critiques of "score-only" vulnerability management are not new. What tends to be missing is a model of how scoring limitations propagate through real remediation workflows. Koscinski et al. study conflicts and inconsistencies across scoring systems and show that the same vulnerability can receive meaningfully different prioritization signals depending on which score you trust. Our contribution is orthogonal: we model the operational reality of patch queues and capacity constraints and show how sensing imperfections (late/missing signals) and execution constraints (backlogs) jointly determine what gets fixed by decision time. In other words, scoring conflicts explain why the signal is confusing; queue dynamics explain why that confusion turns into missed remediation.

### 5.5 Why this matters for mission-constrained environments

Industrial control systems (ICS) make the trade-offs less abstract. In mission-constrained environments, patching competes with safety, uptime, certification, and change-control windows, and "capacity" is often bounded by process, not headcount. Mission-centric approaches to ICS cybersecurity explicitly emphasize operational impact and system function over scalar scoring. Empirically, vulnerability populations in ICS/OT also display strong skew and concentration patterns, with defenders forced to act under harsh constraints. In such settings, relying on a delayed scalar score as a gate is not merely suboptimal; it is an attractive way to formalize blind spots.

# 6 Practical takeaways

EPSS-only thresholding creates a practical trade-off: stricter gates reduce background work but can delay (or miss) exploited items until the score becomes "visible" to the policy. The operational question is not whether EPSS correlates with exploitation, but whether an EPSS-only gate can meet remediation timelines under realistic queueing and capacity constraints.

**Same-day outcome is mostly a *signal* problem at realistic thresholds.** Figure 10 decomposes outcomes at the KEV inclusion date ($t = 0$) for an example capacity of 40 items/week (denominator: all 245 KEV additions in the cohort). Even at the lowest gate shown, only $\approx 28.6\%$ of KEV items are patched by $t = 0$ (70/245). At higher thresholds, the "patched by $t = 0$" share drops further ($\approx 27.3\%$ at $\tau = 0.01$, 67/245; $\approx 20.0\%$ at $\tau = 0.1$, 49/245). Crucially, for $\tau \geq 0.01$ the dominant loss mode is *missing/late eligibility* (i.e., the EPSS signal has not crossed the gate by $t = 0$): $\approx 68.2\%$ (167/245) at $\tau = 0.01$ and $\approx 77.6\%$ (190/245) at $\tau = 0.1$. In contrast, the share missed due to operations queueing at $t = 0$ is small at these higher thresholds ($\approx 4.5\%$ at $\tau = 0.01$, 11/245; $\approx 2.4\%$ at $\tau = 0.1$, $\sim 6/245$). This means that for $\tau \geq 0.01$, increasing staffing/capacity cannot fix same-day KEV response if the gate itself keeps most exploited items ineligible at decision time.

**The 14-day window shows where capacity helps—and where it doesn't.** Figure 9 (zoom $\pm 35$ days around KEV inclusion) shows completion trajectories under background load, with markers at $t = 0$ and the 14-day target. Two patterns stand out. First, at $t = 0$ capacity meaningfully improves completion only under the lowest gate ($\tau = 0.001$), rising from roughly $\sim 28\%$ (cap=40/week) to $\sim 34\%$ (cap=120/week) to $\sim 43\%$ (cap=400/week) in the Figure 9 subsets. For $\tau = 0.01$ and $\tau = 0.1$, the curves at $t = 0$ largely overlap, consistent with Figure 10's finding that the bottleneck is eligibility rather than staffing.

Second, by day 14 the effect of capacity is highly threshold-dependent. For $\tau = 0.001$, 14-day completion rises from $\sim 41\%$ (cap=40/week) to $\sim 49\%$ (cap=120/week) and can approach $\sim 95\%$ at cap=400/week. For $\tau = 0.01$, completion improves from $\sim 72\%$ (cap=40/week) to $\sim 86\%$ (cap=120/week), with little additional gain at cap=400/week. For $\tau = 0.1$, the 14-day outcome saturates around $\sim 56$–$59\%$ across capacities, indicating a regime where added capacity produces minimal improvement because many exploited items remain ineligible for too long under the strict gate.

**Implication.** If you require a "KEV within 14 days" operational target, EPSS-only gating can be workable only in the low-threshold regime (and then becomes capacity-sensitive). In contrast, at $\tau \geq 0.01$ the limiting factor is EPSS timing/availability: capacity increases yield diminishing returns, and achieving the target requires supplementing EPSS with additional decision signals (e.g., a KEV override or other exploitation evidence) rather than simply scaling remediation throughput.

# 7 Limitations

KEV is a conservative but incomplete proxy for exploitation in the wild: it is curated, selective, and may lag or omit real-world exploitation, so our results should be interpreted as bounds relative to a strong public exploitation signal rather than a census of all exploitation. EPSS data availability and timing are also imperfect; we treat missing EPSS as "below threshold" and use weekly snapshots, so "first threshold crossing" should be read as "first observed crossing in our sampling cadence." This

approximates operational visibility in a weekly workflow but does not substitute for fine-grained disclosure timelines (e.g., NVD publish dates) or vendor advisory publication times.

Our queueing model is intentionally simplified to prioritize interpretability: weekly time steps, a fixed per-item handling cost, and FCFS scheduling among eligible items. We interpret per-item effort as security-team handling time (triage, coordination, change control, verification), not raw patch installation time; the 1 hour per vulnerability assumption is a normalization used to convert hours/week into items/week and to express backlog in comparable units across thresholds. Backlog estimates scale linearly with the assumed per-item effort, and alternative scheduling policies (e.g., preemption, asset-aware prioritization, KEV-first overrides) could materially change tail behavior even if decision-time visibility remains constrained by signal availability.

Finally, our focus on the 2025 cohort is deliberate for reproducibility and clarity. Different years, ecosystems, or threat environments may exhibit different base rates, disclosure dynamics, and EPSS behavior; repeating the analysis across multiple years would help quantify the stability of the observed signal-lag dominance at decision time.

# 8 Conclusion

EPSS helps. It does not save you. When used as a gate, EPSS thresholds trade workload for missed exploited vulnerabilities, and the resulting delays are often dominated by signal availability rather than team capacity. In 2025, low thresholds admit large volumes of non-KEV work (e.g., 57.93 non-KEV per KEV at $\tau = 0.001$), while higher thresholds reduce noise but still fail to deliver decision-time completion: for $\tau \geq 0.01$, increasing capacity from 40h/week to 400h/week does not improve KEV completion at $t = 0$, indicating that same-day remediation is constrained by visibility rather than staffing.

The uncomfortable implication is that better outcomes require a better system rather than a single better scalar metric: combine multiple early signals (including but not limited to EPSS), make decision-time reliability targets explicit (e.g., "what fraction of exploited items must be actionable by $t = 0$?"), and manage remediation as a queue-aware process with clear intake rules and overrides. EPSS is valuable as one instrument on the panel; it is insufficient as the only gate controlling which vulnerabilities enter the queue.

## References


[1] J. Jacobs, S. Romanosky, B. Edwards, I. Adjerid, and M. Roytman. Exploit Prediction Scoring System (EPSS). *Digital Threats: Research and Practice*, 2(3), 2021. doi:10.1145/3436242.

[2] FIRST EPSS SIG. FIRST Exploit Prediction Scoring System (EPSS). https://www.first.org/epss/. Accessed 2025-12-30.

[3] FIRST EPSS SIG. Interpreting EPSS probabilities and percentiles. https://www.first.org/epss/articles/prob_percentile_bins. Accessed 2025-12-30.

[4] Cybersecurity and Infrastructure Security Agency (CISA). cisagov/kev-data: Mirror of cisa.gov/kev data files. https://github.com/cisagov/kev-data. Accessed 2025-12-30.

[5] S. Gordeychik, V. Gapanovich, and E. Rozenberg. Signalling cyber security: a mission-centric approach for rail. Railjournal.com, published 2016-07-12. https://www.railjournal.com/in_depth/signalling-cyber-security-the-need-for-a-mission-centric-approach/. Accessed 2025-12-31.

[6] V. Koscinski, M. Nelson, A. Okutan, R. Falso, and M. Mirakhorli. Conflicting Scores, Confusing Signals: An Empirical Study of Vulnerability Scoring Systems. In *Proceedings of the 32nd ACM Conference on Computer and Communications Security (CCS 2025)*, pages 1904–1918, 2025. doi:10.1145/3719027.3765210. Preprint: arXiv:2508.13644.

[7] Kaspersky Lab. ICS Vulnerabilities Statistics (ICS CERT report). https://media.kasperskycontenthub.com/wp-content/uploads/sites/43/2016/07/07190426/KL_REPORT_ICS_Statistic_vulnerabilities.pdf. Accessed 2025-12-31.

## List of Figures

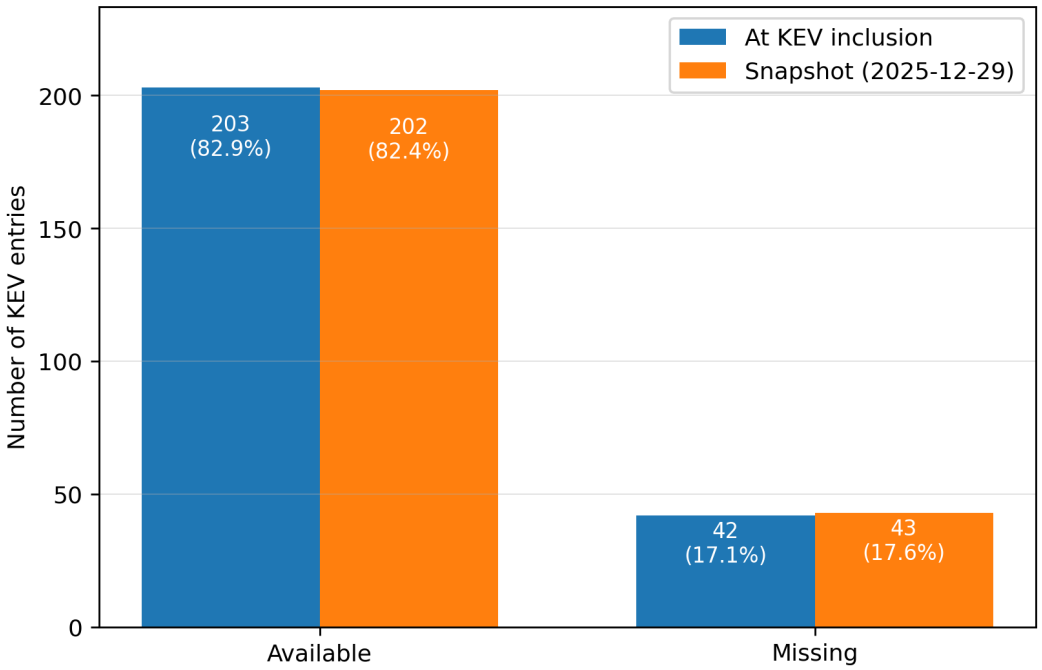


Figure 1: EPSS availability on the KEV `dateAdded` for 2025 additions. Missing scores are operationally equivalent to "below threshold" for an EPSS-only gate.

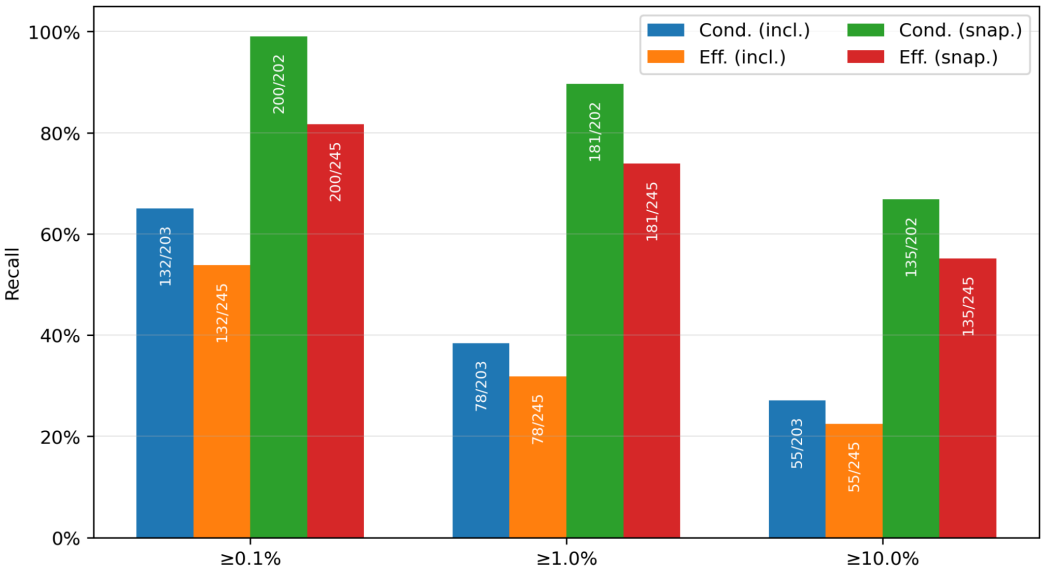


Figure 2: Recall of KEV-at-decision-time under common EPSS thresholds.

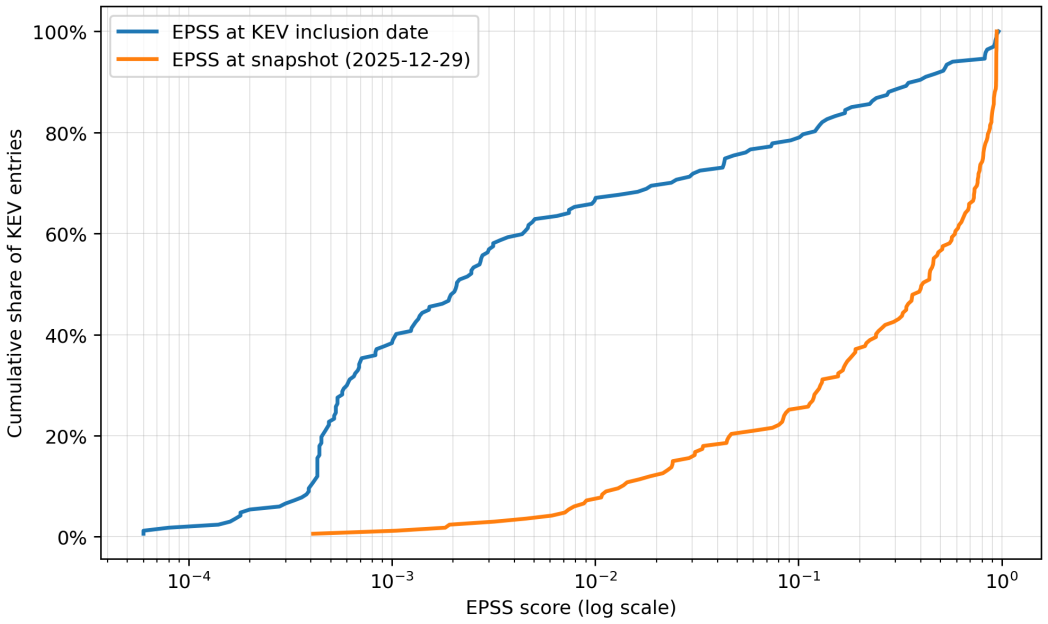


Figure 3: KEV-only ECDF: EPSS at inclusion vs a later snapshot. A significant right-shift indicates that EPSS often becomes larger *after* the KEV decision time.

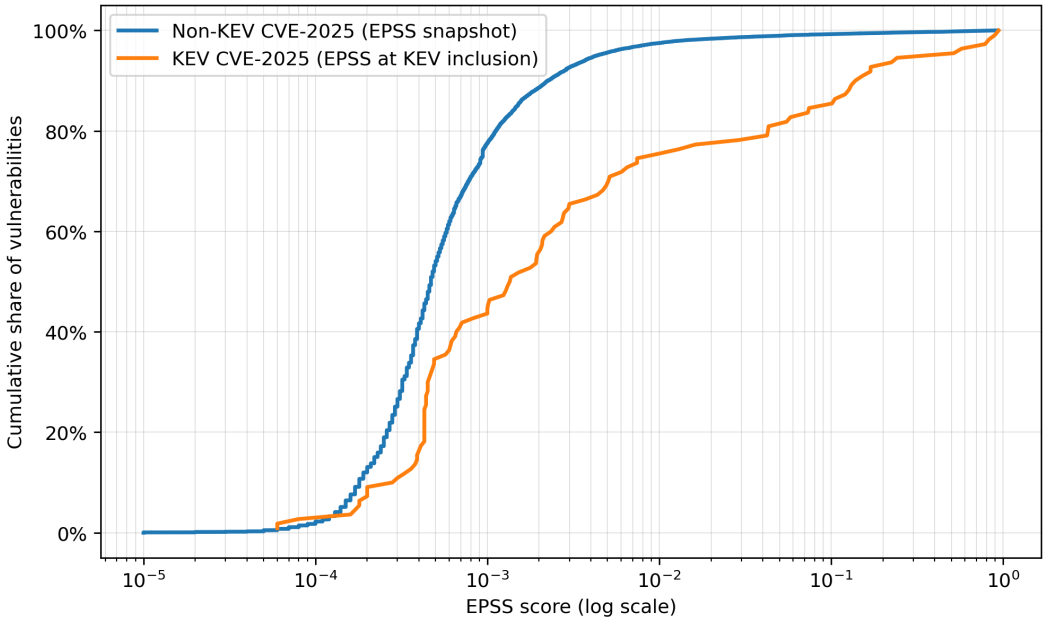


Figure 4: ECDF comparison: KEV (EPSS at `dateAdded`) vs non-KEV (snapshot). Overlap implies that EPSS alone cannot be a perfect separator.

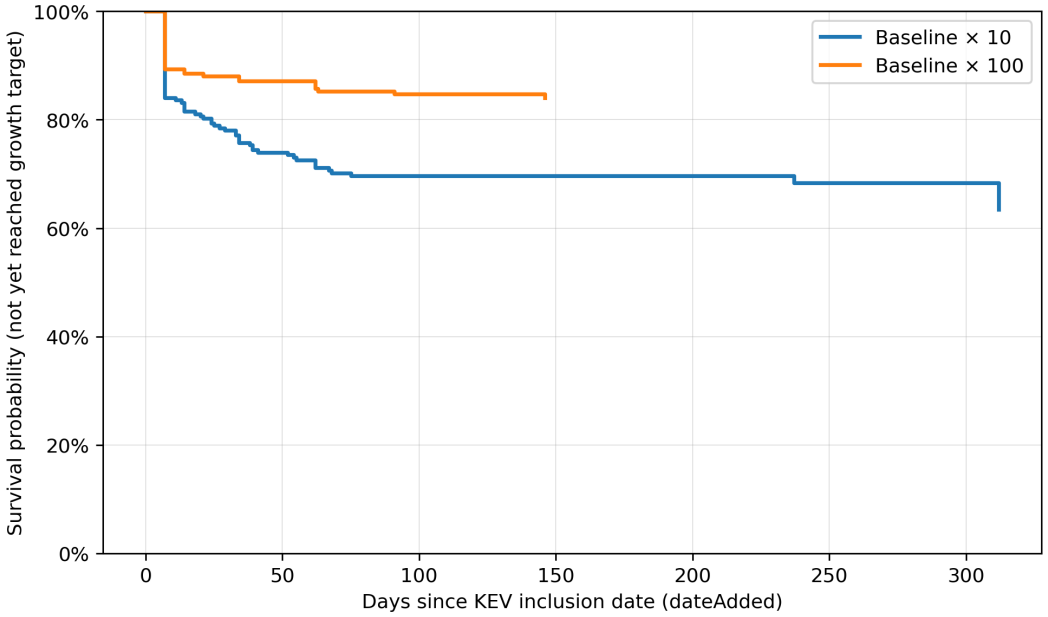


Figure 5: Catch-up latency (Kaplan–Meier) for multiplicative EPSS growth targets: survival probability that EPSS has *not yet* increased by $\times 10$ or $\times 100$ relative to the baseline $e_0$ (first weekly snapshot on/after KEV inclusion).

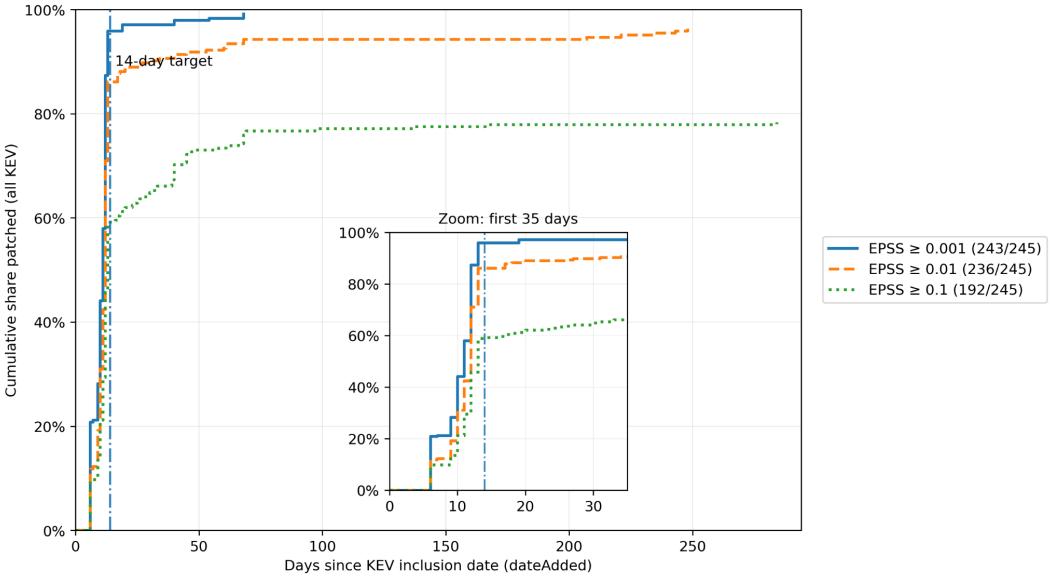


Figure 6: Completion fraction over time under different EPSS thresholds in a simplified weekly closure model.

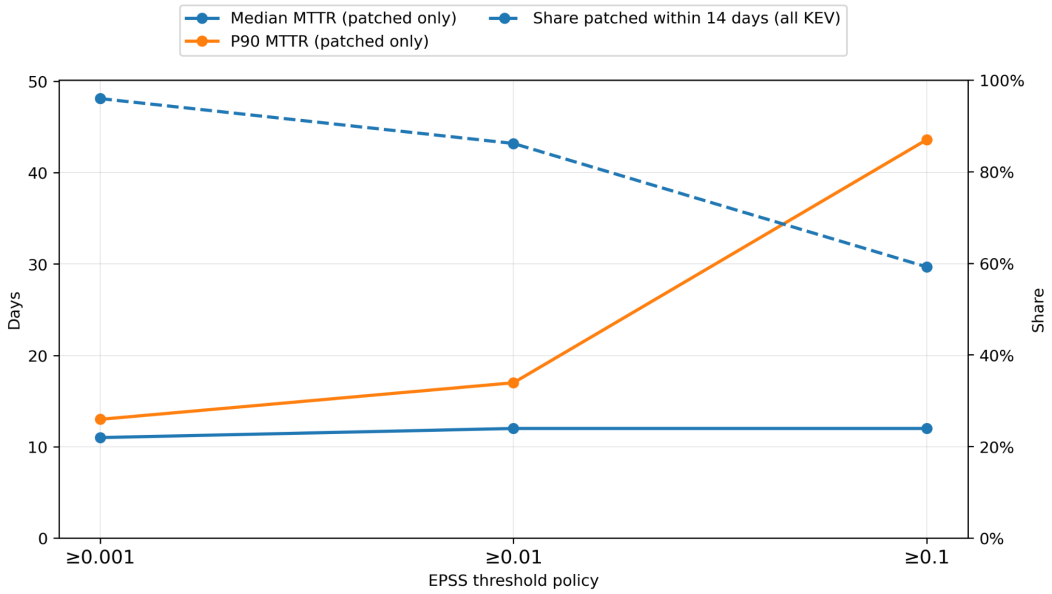


Figure 7: MTTR summary points (median and P90) by threshold. "Pretty" medians can coexist with long tails.

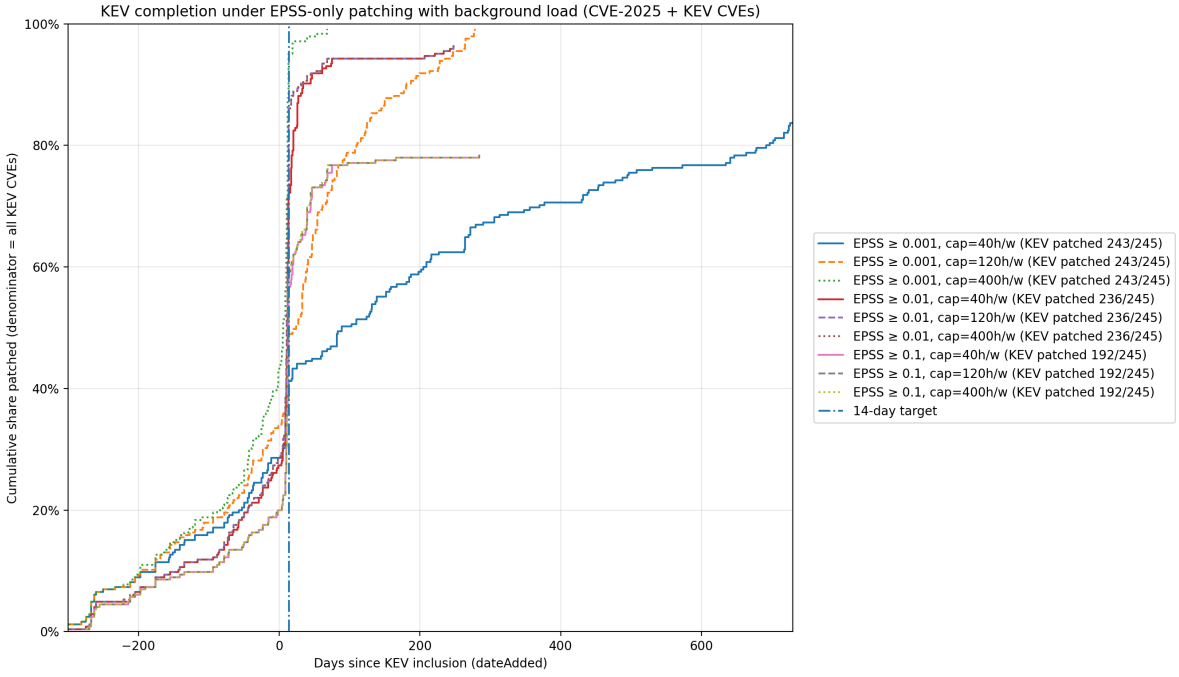


Figure 8: KEV completion curves under nine EPSS-only strategies (threshold × capacity) when the queue includes all eligible CVEs, not just KEV.

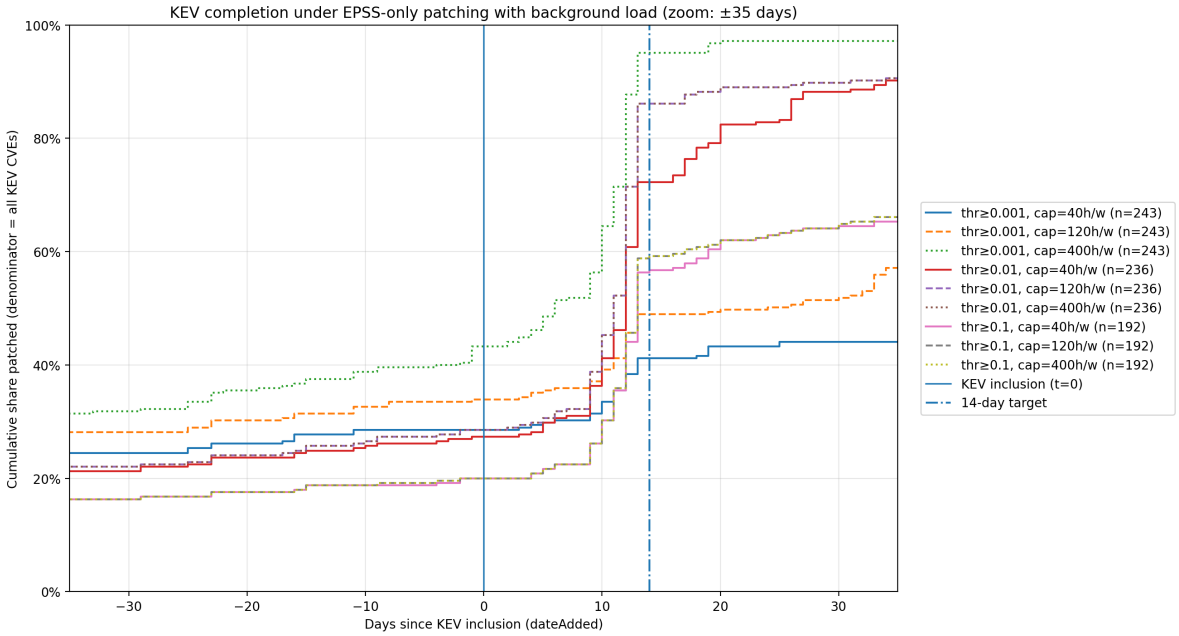


Figure 9: Zoom to ±35 days around KEV inclusion to show the operationally relevant window.

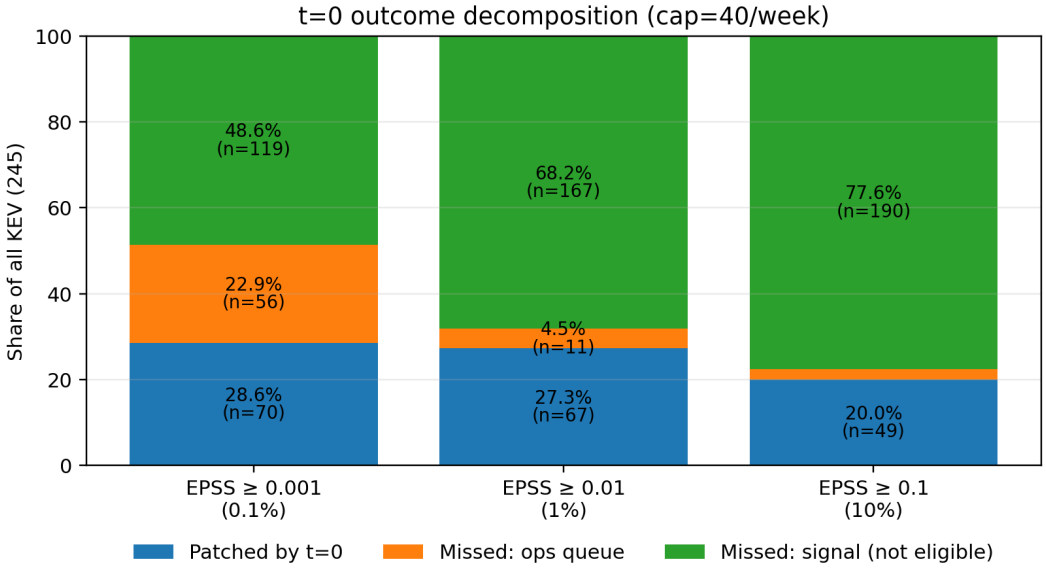


Figure 10: Decomposition at $t = 0$: share of KEV items lost to missing/late EPSS signal vs queued operations (example capacity shown).

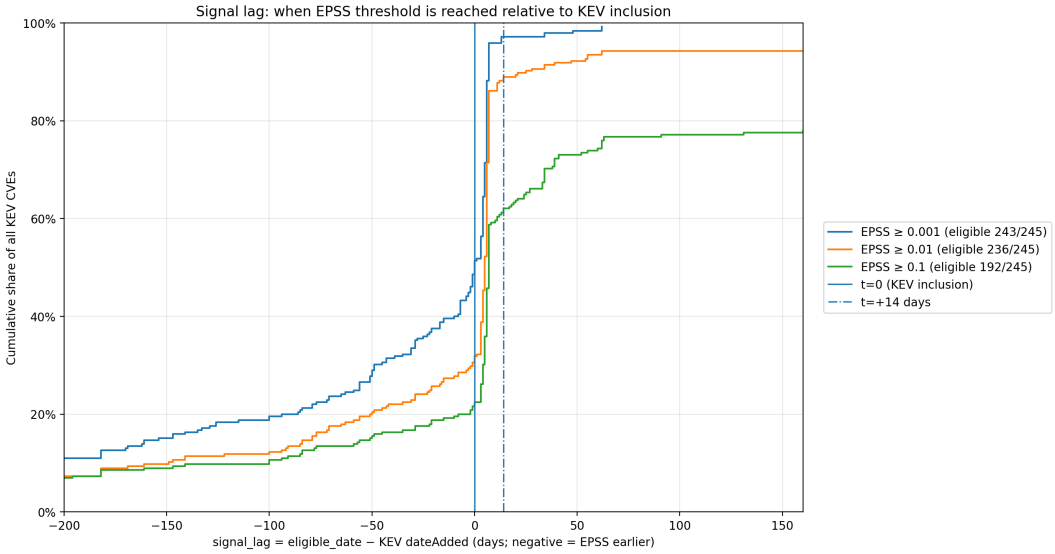


Figure 11: ECDF of signal lag.

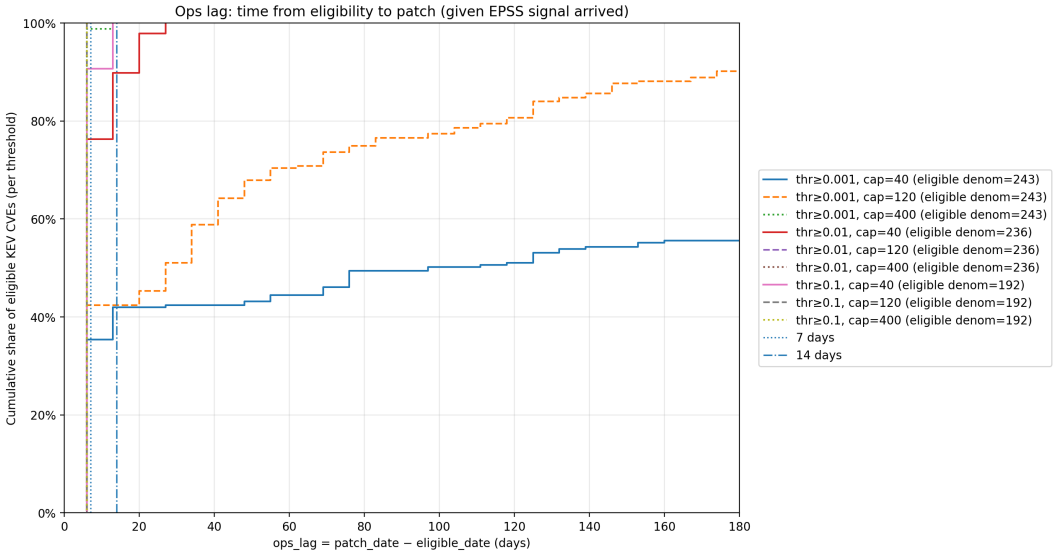


Figure 12: ECDF of operations lag.