UPS Meets Patch Queues
Evidence-timeline prioritization under limited capacity, cadence, and compliance gravity

Sergey Gordeychik
CyberOK Research
textttgordey@cyberok.ru

January 2026

Abstract

Modern vulnerability management must prioritize remediation under extreme disclosure volume and limited patch capacity. We introduce **UPS (Urgent Patch Score)**, an evidence-driven phase model that treats vulnerability urgency as a timeline of verifiable signals rather than a static score. Using the CISA KEV 2025 cohort (N=245) as a high-precision exploited set and a capacity-constrained patch simulation, we show that UPS enables meaningful shift-left remediation without using KEV as a trigger: 35–53% of KEV-2025 items are patched by the KEV addition date. With moderate capacity ($\geq$ 20 items/week), UPS achieves near-perfect 14-day KEV compliance, while tighter deadlines (7 days) are primarily limited by organizational cadence rather than raw capacity. Finally, we quantify the core operational trade-off: earlier admission policies increase workload inflation (an upper-bound overhead factor, defined as total items handled per KEV-2025 item handled; up to $\sim$5–8$\times$ under Watch+), whereas stricter thresholds (Track+/Prepare+) preserve much of the speed benefit at $\sim$2.6–2.8$\times$ overhead. These results suggest that evidence-timeline prioritization can provide a practical, tunable alternative to score-only approaches when capacity and governance constraints dominate outcomes.

Datasets and code used in this study are publicly available at `https://github.com/scadastrangelove/kev_vs_epss`.

1 Introduction

Vulnerability management operates under finite remediation throughput and a large, continuously growing disclosure volume. Patch execution is bounded by engineering time, testing, change windows, and governance constraints.

A range of scoring and decision frameworks exist to assist prioritization: CVSS for technical severity, EPSS for exploitation probability, plus assorted vendor scores and decision trees. However, operational outcomes depend on queue dynamics under capacity and cadence constraints, not on a score in isolation. The operational question is not "how scary is this CVE," but:

> **Given our capacity and cadence, what will we patch on time—and what will we miss?**

This is naturally a *queueing* problem: priorities only matter through scheduling under finite capacity and discrete planning cadence.

This paper takes an operational approach:

1. Treat urgency as a **timeline of evidence**, not a scalar number.
2. Convert the timeline into **phase-based priorities**.
3. Run it through a **capacity-bounded patch queue** with a real calendar.
4. Measure outcomes against **external compliance clocks**, not internal optimism.

Contributions.

- We define UPS, an evidence-timeline phase model that maps timestamped public artifacts to monotone operational urgency stages.
- We evaluate UPS under a capacity-bounded, weekly-cadence patch queue on the CISA KEV 2025 cohort, quantifying shift-left remediation and absolute-SLA outcomes.
- We quantify the speed–workload trade-off introduced by earlier admission thresholds using an upper-bound workload inflation factor.
- We provide an extension that incorporates EPSS and public-attention sightings as bounded soft signals under explicit load budgets.
- We discuss how UPS phases can be used as evidence inputs to stakeholder-specific decision workflows (e.g., SSVC).[22]

2 Related Work

2.1 Scoring systems optimize different objectives

Koscinski et al. show empirically (and outcome-linked) that different scoring systems are *not interchangeable* and can disagree sharply on what is "top priority."[2] This is not shocking: systems optimize different objectives, use different data, and carry different institutional incentives. Accordingly, a single score should not be treated as a universal prioritization oracle.

2.2 EPSS-only prioritization meets the patch queue

Prediction is useful; assumptions are dangerous. In *Prediction Meets Patch Queues*, we evaluate the operational pattern "patch EPSS$> \tau$" under capacity constraints and show systematic limits when judged against exploited ground truth as represented by KEV additions in 2025.[1] That work motivates the next step: if score-only gates fail under queue dynamics, perhaps we should stop forcing a scalar to do the job of a timeline.

2.3 Context-aware prioritization and decision frameworks

Recent work has proposed integrating exploitability estimates with business context, workflow constraints, and decision support. Examples include business-aligned risk translation models such as RiskBridge, [13] chaining-style workflow frameworks, [14] and context-aware patch prioritization models. [15, 16] Survey work summarizes exploitability assessment approaches and motivates using multiple signal types rather than any single score or signal in isolation. [21]

Table 1: Priority stability mechanisms in UPS.

Mechanism	Operational effect
Sticky stage (monotone)	Prevents "today urgent, tomorrow meh" oscillation; once escalated, items do not de-escalate without explicit policy.
Floor stage (high-precision evidence)	High-confidence signals (e.g., exploited-catalog additions) set a non-negotiable minimum.
Policy floor (admission threshold)	Tunable noise filter: Watch+/Track+/Prepare+ define how early you are willing to spend effort.
Event auditability	Each transition is tied to a timestamped artifact; supports post-incident analysis with traceable evidence.

3 UPS: urgency as a timeline of verifiable signals

UPS (Urgent Patch Score) is not a "better number." It is a **phase machine** driven by timestamped evidence.

3.1 Operational phases

UPS models a vulnerability's lifecycle through phases such as:

- **Radar**: "exists" but not yet operationally actionable.
- **Watch**: early public signals justify lightweight attention.
- **Track**: stronger evidence suggests rising exploitation likelihood.
- **Prepare**: practical readiness work should be underway.
- **Urgent Patch / Emergency**: patch immediately under emergency change control.

The labels map to actions (triage depth, detection effort, change planning), not just risk vocabulary.

3.2 Evidence signals

UPS consumes public artifacts as evidence events: *(type, timestamp, evidence)*. Examples include advisories, PoCs, exploit modules/templates, detection rules, exploitation reporting, and exploited catalogs such as CISA KEV.[7] A key design choice is that UPS does not require *all* signals to exist. It only requires that *some* signals exist early enough, often enough, to move work left.

3.3 UPS phase machine and stability rules

UPS is deliberately conservative about priority oscillation. Two principles dominate: (i) evidence can escalate urgency, and (ii) urgency should not de-escalate without explicit policy.

Priority stability mechanisms. Table 1 summarizes the rules that prevent oscillation and make "why was this urgent?" answerable.

State update rule. Given a stream of events sorted by timestamp, UPS computes an instantaneous stage suggestion (`stage_point`), a non-decreasing floor stage derived from high-precision evidence (`floor_stage`), a policy minimum stage (`policy_min`), and a monotone aggregate stage (`stage_sticky`). The result is a phase timeline suitable for priority scheduling.

Listing 1: UPS phase update (pseudocode).

```
# Inputs:
# events: list of evidence events, each with (ts, type, fields)
# policy_min: admission threshold (Watch+/Track+/Prepare+)
# map_stage_point(type): maps event type -> instantaneous suggested stage
# map_floor(type): maps high-precision types -> floor stage (or None)

stage_sticky = RADAR # evidence-derived, monotone
floor_stage = RADAR # evidence-derived, monotone
admitted = False
admit_ts = None

for e in sort_by_ts(events):
    # 1) instantaneous suggestion from THIS event only
    stage_point = map_stage_point(e.type)

    # 2) update high-precision floor (if applicable)
    f = map_floor(e.type)
    if f is not None:
        floor_stage = max(floor_stage, f)

    # 3) evidence-derived urgency (monotone, independent of policy)
    stage_sticky = max(stage_sticky, stage_point, floor_stage)

    # 4) policy gates admission; it does not rewrite evidence state
    if (not admitted) and stage_sticky >= policy_min:
        admitted = True
        admit_ts = e.ts # optionally: next_planning_tick(e.ts)

    # optional: append raw evidence artifacts to an audit ledger (independent of
        admission)
    # record_evidence(e)

    # emit phase timeline only once the item is admitted (policy-gated)
    if admitted:
        emit(ts=e.ts,
             stage_point=stage_point,
             floor_stage=floor_stage,
             stage_sticky=stage_sticky,
             admitted=admitted,
             admit_ts=admit_ts)
```

4 Data

4.1 UPS timelines dataset

Our UPS signal collection yields **6,312 unique CVEs** (1999–2026) with at least one recorded event in `timelines.csv`. Of these, **5,492** enter an actionable (non-Radar) phase at least once.

For the year 2025, UPS highlights a small actionable subset:

- Watch+: 752 CVE-2025 items
- Track+: 536 CVE-2025 items

- Prepare+: 500 CVE-2025 items

This reflects the core premise: most CVEs do not receive strong external evidence signals and should not dominate scarce triage/patch capacity.

4.2 CISA KEV 2025 as high-precision exploited set

We use the CISA Known Exploited Vulnerabilities (KEV) catalog additions in 2025 (N=245) as a high-precision exploited cohort.[7] KEV is not perfect ground truth, but it is one of the clearest public "exploited means exploited" labels available.

We use *future-KEV* as a time-relative label: for a KEV-2025 CVE with KEV addition time t_{KEV}, at any earlier time $t < t_{\text{KEV}}$ the item is *future-KEV* (known by open signals but not yet KEV-listed). Shift-left is work completed by t_{KEV} without treating KEV itself as a trigger.

4.3 EPSS snapshots for 2025

We use weekly EPSS probability and percentile snapshots for the 2025 evaluation window.[11, 12] EPSS is used only as an auxiliary soft signal in the extension analysis; it is not treated as a trigger for operational work in the core UPS queue evaluation.

4.4 vKEV as a sensitivity check

We additionally analyze VulnCheck KEV ("vKEV") as an alternative exploited-catalog timeline.[10] It is not a federal compliance standard in the way CISA KEV is, but it is useful for a robustness check on lead time and on "pre-catalog" evidence readiness.

5 Methods: patch queue simulation under capacity and cadence constraints

5.1 Queue model

We simulate a discrete-time patch process:

- weekly cadence (planning tick),
- service capacity of 10/20/40 items per week,
- admission policies: Watch+, Track+, Prepare+ (policy minimum stage),
- scheduling by phase priority with escalation (Emergency > Urgent > Prepare > Track > Watch),
- constant service time per work item.

Service time is modeled as constant to isolate the effect of evidence timing, finite capacity, and planning cadence. In deployment, heterogeneous effort, bundling, dependencies, and asset exposure filtering will further shape outcomes.

5.2 Absolute and relative SLA definitions

We report two SLA semantics to reflect common compliance clocks and internal process clocks. **Absolute SLA** is calendar-anchored: deadlines are measured from an outside timestamp (here, KEV `dateAdded`) and do not care about internal ticket creation rituals.[8] Throughout, we use fixed 7-day and 14-day deadlines as *hypothetical internal SLAs* for comparability; they should not be read as due-date compliance, since catalog due dates can vary across entries. **Relative SLA** is the internal process view: deadlines are measured from discovery / scan time / ticket creation—a common framing in audit-driven workflows (e.g., patching critical vulnerabilities within 30 days in PCI DSS language).[9] Under a weekly planning cadence, absolute and relative deadlines can diverge systematically.

5.3 Cadence penalty and the "hotfix lane" upper bound

Weekly cadence introduces delay between an external event and the next planning tick. We quantify this as *cadence penalty* (0–6 days). We also compute a bounded "hotfix lane" upper bound: if urgent items can preempt mid-cycle, how much of strict absolute SLA failure is cadence-limited rather than capacity-limited?

6 Results

We evaluate nine scenarios (capacity × policy) on the KEV-2025 cohort. We report multiple complementary views to separate signal timing effects from operational timing effects.

6.1 Shift-left: patching future-KEV before KEV arrives

UPS enables meaningful shift-left remediation without using KEV as a trigger: 35–53% of KEV-2025 items are already patched by the KEV addition date (day0), depending on policy and capacity (Supplementary Figure S1; completion-time distribution in Supplementary Figure S2). Interpretation: the system does not need KEV to start moving. It needs enough early signals to justify admission and preparation for a subset that later becomes KEV.

6.2 Signal latency vs ops latency: diagnose before prescribing

If you are late, first decide whether you were late to know (signal lag) or late to act (ops lag). Supplementary Figure S3 makes this separation explicit.

6.3 7-day absolute compliance is often a cadence problem

For tight deadlines (7 days), cadence becomes the dominant limiter (Supplementary Figure S4). A controlled exception workflow (a "hotfix lane") can reduce cadence-induced delay, but it consumes separate budget and governance capacity.

6.4 Completion envelope visualization

We visualize patch completions relative to KEV `dateAdded` using a kernel-smoothed completion density estimate. This representation highlights whether remediation concentrates at the external trigger time or is distributed earlier via pre-KEV admission and preparation.

Table 2: KEV share and workload inflation in 2025 (upper bound; KEV is incomplete).

Cap/wk	Policy	Admitted	KEV share in admitted (%)	Overhead factor	Patched	KEV share in patched (%)	Overhead factor
10	Prepare+	520	39.2	2.55×	481	41.8	2.39×
10	Track+	551	36.7	2.73×	482	41.1	2.43×
10	Watch+	1388	13.0	7.67×	520	35.6	2.81×
20	Prepare+	520	39.2	2.55×	520	39.2	2.55×
20	Track+	551	36.7	2.73×	551	36.7	2.73×
20	Watch+	1388	13.0	7.67×	989	18.3	5.46×
40	Prepare+	520	39.2	2.55×	520	39.2	2.55×
40	Track+	551	36.7	2.73×	551	36.7	2.73×
40	Watch+	1388	13.0	7.67×	1366	13.3	7.55×

Table 3: Scenario summary (operating points).

Cap/wk	Policy	Admission overhead factor	Abs SLA@14d (%)	Hotfix pressure @7d (%)
10	Prepare+	2.55×	97.1	34.3
10	Track+	2.73×	97.6	33.9
10	Watch+	7.67×	98.0	30.6
20	Prepare+	2.55×	100.0	33.9
20	Track+	2.73×	100.0	32.2
20	Watch+	7.67×	100.0	29.0
40	Prepare+	2.55×	100.0	33.9
40	Track+	2.73×	100.0	32.2
40	Watch+	7.67×	100.0	28.2

UPS policies shift work left: visible pre-KEV activity reduces dependence on a single exploited-catalog trigger (Supplementary Figure S5). The remaining post-KEV mass reflects remediation that could not be completed safely before KEV addition.

7 Workload inflation and the price of early admission

Shift-left remediation is not free. If KEV is treated as exploited ground truth, then work performed on non-KEV items looks like overhead. KEV is incomplete, so this is an upper bound; nevertheless, it is the operational budget you will be asked to justify.

Definition (upper-bound overhead). Let $\mathcal{W}$ be the set of work items admitted (or patched) under a scenario in 2025, and let $\mathcal{K} \subseteq \mathcal{W}$ be those items that are in KEV-2025. We report the *KEV share* $|\mathcal{K}|/|\mathcal{W}|$ and its inverse as an upper-bound *workload inflation/overhead factor*: $|\mathcal{W}|/|\mathcal{K}|$ (i.e., total items handled per KEV-2025 item handled). This is intentionally *not* normalized by the full KEV-2025 cohort size (245), because scenarios may fail to admit/patch a fraction of KEV items by the measurement time.

Supplementary Figure S6 shows the non-KEV share of admitted and patched work under each scenario (an upper bound on overhead because KEV is incomplete).

8 Soft-signal extensions: EPSS and public attention

UPS can incorporate soft signals as bounded admission hints, provided that induced workload is treated as an explicit budget rather than an implicit externality. Two practically useful channels are EPSS and aggregated public attention.

Table 4: Selected EPSS operating points (Urgent Patch floor, 2025 window; N_{pos}=308). Recall and precision are for early alerts; FP/pos is false positives per positive (extra-load ratio). Lead time is measured in days.

Policy	Rule	Recall	Precision	FP/pos	Lead time (median; p10; p90)
WATCH+	$p \geq 0.30$, $k = 1$	0.1920	0.1180	1.4290	21.00; 2.82; 122.60
TRACK+	$p \geq 0.85$, $k = 2$	0.0747	0.1345	0.4810	15.82; 3.20; 145.10
PREPARE+	$p \geq 0.95$, $k = 1$	0.0552	0.2429	0.1720	18.91; 1.49; 107.65
PREPARE_ACCEL+	$p \geq 0.95$, $\Delta p \geq 0.02$, $k = 1$	0.0292	0.2647	0.0812	21.00; 5.27; 169.30

8.1 EPSS as an admission hint under explicit load budgets

EPSS provides weekly probability and percentile estimates for many CVEs. [11, 12] We define an "early alert" as an EPSS signal that occurs strictly before the vulnerability reaches the operational Urgent Patch floor stage (t_+) within the window. This yields a transparent trade-off: earlier recall increases induced workload, which we report as false positives per positive (FP/pos, an extra-load ratio).

Supplementary Figure S9 shows the Pareto surface (early recall vs FP/pos) and the selected operating points. Supplementary Figure S10 illustrates a ceiling effect: EPSS availability is not universal and may lag relative to the operational positive time.

8.2 Public attention sightings as bounded preparation accelerators

Aggregated sightings from news and community sources can provide limited early warning for a subset of late-knowledge cases, but these signals are noisy and biased. [18–20] Using CIRCL's Vulnerability-Lookup sightings API, [17] simple boundary rules can trigger low-cost preparation work (scoping, exposure checks, detection readiness) under an explicit cap: attention alone should not raise the stage beyond Track unless corroborated by harder evidence (e.g., PoCs, exploit modules, vendor "confirmed in-the-wild" statements).

9 Discussion

UPS relies on publicly observable artifacts (advisories, exploit artifacts, detection content, and exploited catalogs) as timestamped evidence. This section discusses why such artifacts are produced at scale and how they can be used responsibly despite bias and strategic behavior.

9.1 Incentives for public artifact production

In markets with information asymmetry, observable signals are used to demonstrate competence and responsiveness.[3] In vulnerability response, these signals take the form of advisories, technical analyses, mitigations, and detection guidance.

Table 5: Example attention boundaries using CIRCL sightings (late-knowledge KEV slice; N=41). Shares and lead times are reported relative to KEV `dateAdded`.

Boundary	Triggered (count)	Share	Lead time (days)
Any sightings	10	24%	informational
Any non-social sightings	6	15%	informational
Trusted non-social (A)	5	12%	median ∼0.8; max 18
Diversified non-social (B)	3	7%	median ∼0.39; max 18
Buzz threshold (C)	2	5%	median ∼10.6 ($n = 2$); max 18

9.2 Disclosure incentives and externalities

Disclosure and coordination mechanisms create externalities across vendors, researchers, coordinators, and defenders. Prior work frames disclosure as an equilibrium shaped by incentives, liability, and coordination costs.[6]

9.3 Economics of information sharing

Economic models show conditions where security-relevant information sharing can be individually rational and can reduce the tendency to defer security investments.[4, 5] Many defensive artifacts therefore function as information-sharing events even when they are disseminated through commercial channels.

9.4 Amplification of weak signals

Public artifacts often propagate through toolchains: an advisory may lead to PoCs, scanning templates, and detection rules. UPS is designed to exploit this propagation by treating artifacts as evidence for phase transitions rather than as ground truth.

9.5 Caveats: bias and strategic noise in OSINT

Public attention signals (e.g., social media) are noisy and biased.[18–20] For this reason, the soft-signal extensions explicitly bound their influence and treat induced workload as a budgeted cost.

9.6 Future work

Promising directions include modeling heterogeneous service times and bundling, integrating asset exposure filtering, evaluating additional exploited-catalog timelines, and mapping evidence-timeline phases to stakeholder-specific decision frameworks such as SSVC.[22]

Table 6: Pre-vKEV admission rates in 2025 while excluding `VKEV_ADDED` as an evidence signal (N=888). "Weekly tick" applies the next planning boundary after the pre-vKEV evidence time.

Threshold	Pre-vKEV evidence	%	Admitted by weekly tick	%
Watch+	399	44.9	386	43.5
Track+	181	20.4	167	18.8
Prepare+	172	19.4	159	17.9

10 vKEV as a sensitivity check

We treat vKEV (VulnCheck's exploited catalog) as a robustness check, not as a compliance clock. We assess whether the UPS evidence stream carries actionable signal *before* an exploited-catalog label arrives, and compare exploited-catalog timelines.

10.1 Pre-vKEV evidence readiness (excluding vKEV as a signal)

For vKEV additions in 2025 (N=888), we compute the UPS phase immediately *before* the vKEV listing timestamp while excluding the `VKEV_ADDED` event from the evidence stream. This is the "no answer key" check: how far the pipeline progresses on open signals alone. Stage distribution is shown in Supplementary Figure S7.

10.2 Pre-admission rates under Watch+/Track+/Prepare+

We also measure whether each vKEV-added item would have been admitted *before* vKEV under the three admission thresholds, both in calendar time and after applying a weekly planning tick. The headline is that open signals provide non-trivial early readiness for a large minority under Watch+, and a smaller but meaningful fraction under Track+/Prepare+.

10.3 Lead time of exploited catalogs: vKEV vs CISA KEV

For CVEs appearing in both catalogs in 2025 (N=196 overlap), we compute the distribution of (CISA KEV `dateAdded` − vKEV `dateAdded`). Positive values mean vKEV is earlier. Median lead is modest (~2 days), but the tail is wide—the market does not synchronize its bragging rights (Supplementary Figure S8).

11 Limitations

- **No asset context (universal presence).** We simulate queue dynamics conditional on a CVE being in-scope; we do not model whether a specific organization is actually exposed to each CVE. Therefore absolute admitted/patched volumes are upper bounds; in deployment, inventory/exposure filtering shrinks both workload and the set of KEV items in scope, and shift-left should be measured on the in-scope subset.
- **Constant service time.** Patch effort is heterogeneous; bundling exists; dependencies exist. We model one unit per item to isolate signal + capacity effects.
- **Catalog bias.** KEV is incomplete ground truth; "overhead vs KEV" is an upper bound.

- **Cadence is simplified.** Weekly batching is common, but real organizations have mixed cadences and exception workflows.

12 Conclusion

UPS reframes vulnerability urgency as an evidence timeline and evaluates it where it matters: a capacity-bounded queue subject to compliance clocks. On KEV-2025, UPS enables meaningful shift-left remediation without using KEV as a trigger, reaches near-perfect 14-day absolute compliance at moderate capacity, and makes the speed-versus-workload trade-off explicit. Operational outcomes are determined by queue dynamics under capacity and cadence constraints rather than by any single score in isolation.

Summary of results.

- UPS enables measurable shift-left remediation without treating KEV as a trigger.
- For 14-day external deadlines, moderate capacity is sufficient; for 7-day deadlines, cadence often dominates, motivating a controlled exception workflow.
- Earlier admission increases workload inflation; Track+/Prepare+ retain much of the speed benefit at substantially lower overhead than Watch+.
- UPS phases can be mapped to stakeholder-specific decision workflows (e.g., SSVC) as evidence-driven inputs rather than as a replacement for organizational context.[22]

13 Data and code availability

Datasets and code used in this study are publicly available at `https://github.com/scadastrangelove/kev_vs_epss`. The repository includes the public timeline dataset (`timelines.csv`) and the UPS signal taxonomy used in this work.

References

References

[1] Sergey Gordeychik. *Prediction Meets Patch Queues: Empirical Limits of EPSS-Only Prioritization Using CISA KEV Additions in 2025.* TechRxiv preprint, 2026. DOI: 10.36227/techrxiv.176857939.95987957/v1.

[2] Viktoria Koscinski, Mark Nelson, Ahmet Okutan, Robert Falso, and Mehdi Mirakhorli. *Conflicting Scores, Confusing Signals: An Empirical Study of Vulnerability Scoring Systems.* arXiv:2508.13644, 2025.

[3] Michael Spence. Job market signaling. *The Quarterly Journal of Economics*, 87(3):355–374, 1973. DOI: 10.2307/1882010.

[4] Esther Gal-Or and Anindya Ghose. The economic incentives for sharing security information. *Information Systems Research*, 16(2):186–208, 2005. DOI: 10.1287/isre.1050.0053.

[5] Lawrence A. Gordon, Martin P. Loeb, William Lucyshyn, and Lei Zhou. The impact of information sharing on cybersecurity underinvestment: A real options perspective. *Journal of Accounting and Public Policy*, 34(5):509–519, 2015. DOI: 10.1016/j.jaccpubpol.2015.05.001.

[6] ENISA. *Economics of Vulnerability Disclosure.* European Union Agency for Cybersecurity, December 2018.

[7] CISA. Known exploited vulnerabilities catalog. U.S. Cybersecurity and Infrastructure Security Agency.

[8] CISA. Binding Operational Directive 22-01: Reducing the Significant Risk of Known Exploited Vulnerabilities. U.S. Cybersecurity and Infrastructure Security Agency, November 2021.

[9] PCI Security Standards Council. *Payment Card Industry Data Security Standard (PCI DSS) v4.0.1.* June 2024.

[10] VulnCheck. VulnCheck KEV (Known Exploited Vulnerabilities): Documentation. VulnCheck Community Docs.

[11] FIRST. Exploit Prediction Scoring System (EPSS). `https://www.first.org/epss/`. Accessed 2026-01-31.

[12] FIRST. EPSS data and statistics (downloadable scores and percentiles). `https://www.first.org/epss/data_stats`. Accessed 2026-01-31.

[13] Yelena Mujibur Sheikh, et al. RiskBridge: Turning CVEs into Business-Aligned Patch Priorities. arXiv:2601.06201, 2026.

[14] Naoyuki Shimizu and Masaki Hashimoto. Vulnerability Management Chaining: An Integrated Framework for Efficient Cybersecurity Risk Prioritization. arXiv:2506.01220, 2025.

[15] Abdulrahman Alhomidi, Mark Reed, and Stewart Kowalski. CAVP: A context-aware vulnerability prioritization model. *Computers & Security*, 116:102628, 2022. doi: 10.1016/j.cose.2022.102628.

[16] Adeel Zaidi, et al. SmartPatch: A patch prioritization framework. *Computers in Industry*, 137:103595, 2022. doi: 10.1016/j.compind.2021.103595.

[17] CIRCL. Vulnerability-Lookup: Sightings API. `https://vulnerability.circl.lu/sightings/`. Accessed 2026-01-31.

[18] Carl Sabottke, Octavian Suciu, and Tudor Dumitras. Vulnerability Disclosure in the Age of Social Media: Exploiting Twitter for Predicting Real-World Exploits. In *24th USENIX Security Symposium (USENIX Security 15)*, 2015. `https://www.usenix.org/conference/usenixsecurity15/technical-sessions/presentation/sabottke`.

[19] Haipeng Chen, et al. Using Twitter to Predict When Vulnerabilities will be Exploited. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '19)*, 2019. doi: 10.1145/3292500.3330742.

[20] Prasha Shrestha, et al. Multiple social platforms reveal actionable signals for software vulnerability awareness: A study of GitHub, Twitter and Reddit. *PLOS ONE*, 15(3):e0230250, 2020. doi: 10.1371/journal.pone.0230250.

[21] Sarah Elder, et al. A Survey on Software Vulnerability Exploitability Assessment. *ACM Computing Surveys*, 56(8), 2024. doi: 10.1145/3648610.

[22] Jonathan Spring, Allen D. Householder, Eric Hatleback, Art Manion, Madison Oliver, Vijay S. Sarvepalli, Laurie Tyzenhaus, and Charles G. Yarbrough. Prioritizing Vulnerability Response: A Stakeholder-Specific Vulnerability Categorization (Version 2.0). Software Engineering Institute, Carnegie Mellon University, April 30, 2021. `https://www.sei.cmu.edu/library/prioritizing-vulnerability-response-a-stakeholder-specific-vulnerability-categorization-version-20/`.

Supplementary figures

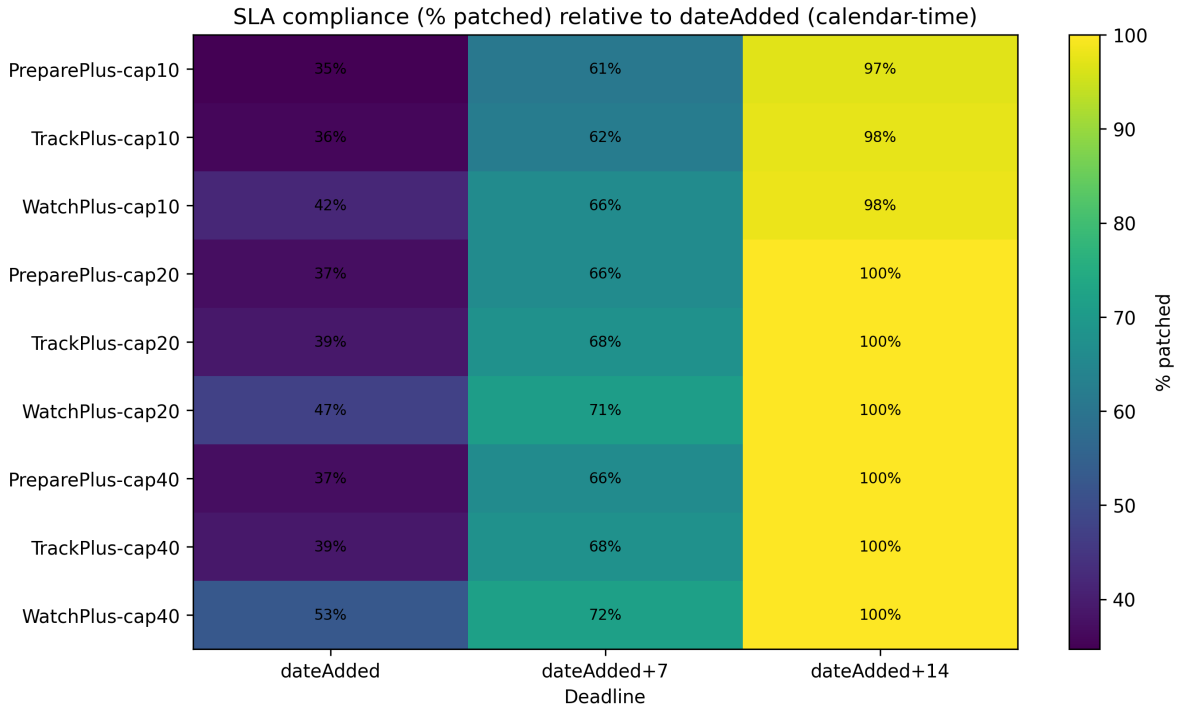

Figure S1: Pre-KEV patching rates (KEV-2025). Heatmap of the fraction of KEV-2025 items patched by (i) the KEV addition date (negative/zero lateness), (ii) within 7 days, and (iii) within 14 days, under nine scenarios (capacity × admission policy).

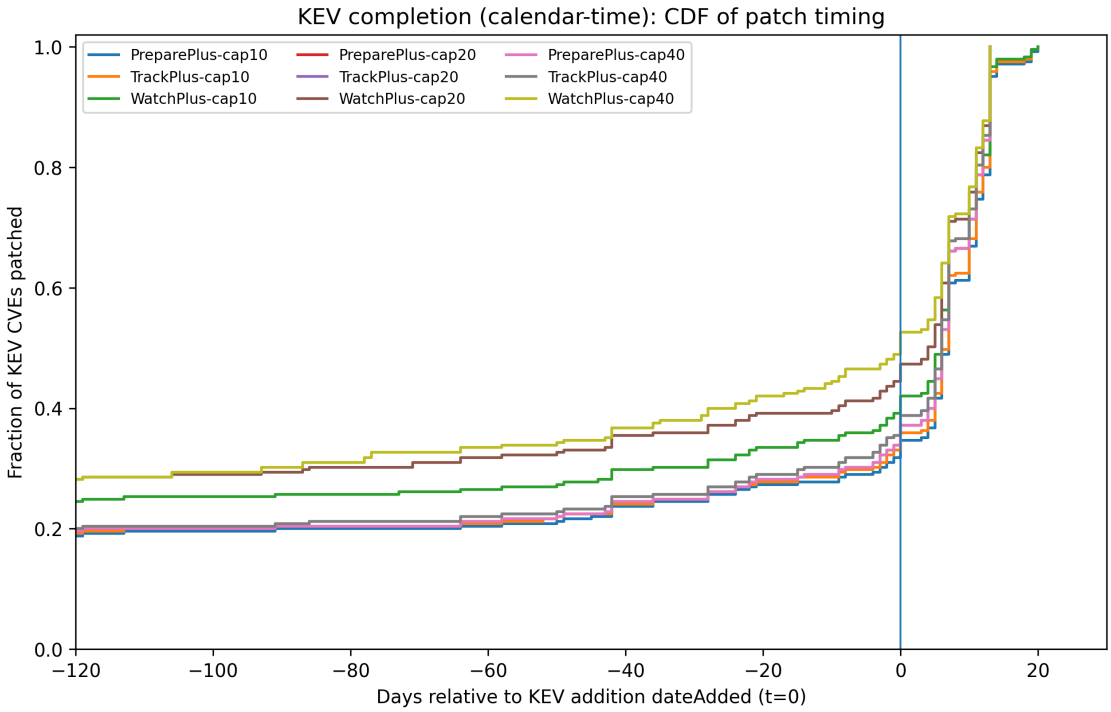

Figure S2: Completion ECDF relative to KEV `dateAdded`. Empirical CDF of completion time offsets for KEV-2025 across scenarios; values below 0 represent shift-left remediation.

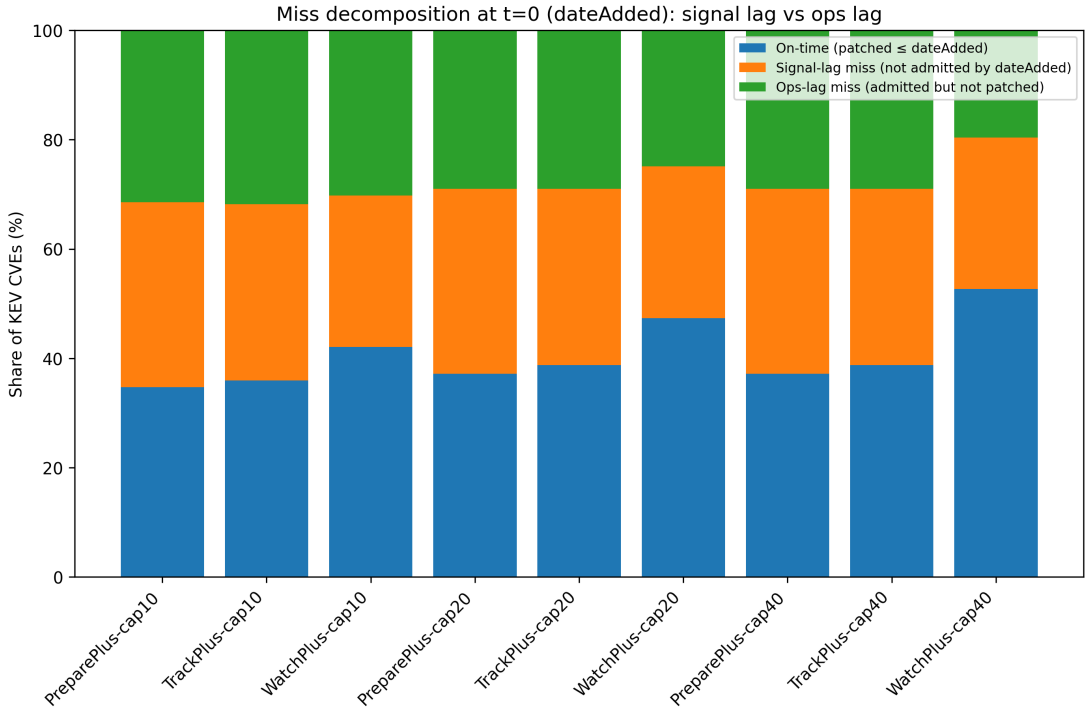

Figure S3: Miss decomposition: signal latency vs ops latency. For KEV-2025, we separate lateness into (i) evidence/signal lag ("late to know") and (ii) operational/cadence lag ("late to act").

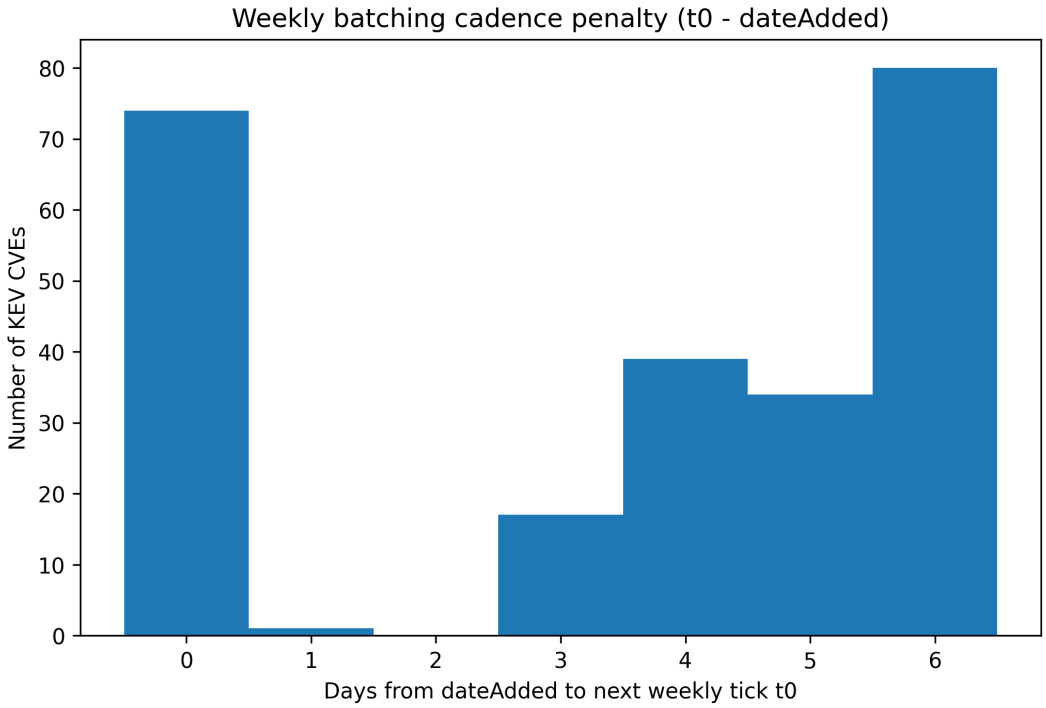

Figure S4: Cadence penalty and hotfix lane upper bound. The weekly planning tick introduces a 0–6 day delay between an external event and actionable admission; a bounded hotfix lane estimates how much of 7-day absolute SLA failure is cadence-limited rather than capacity-limited.

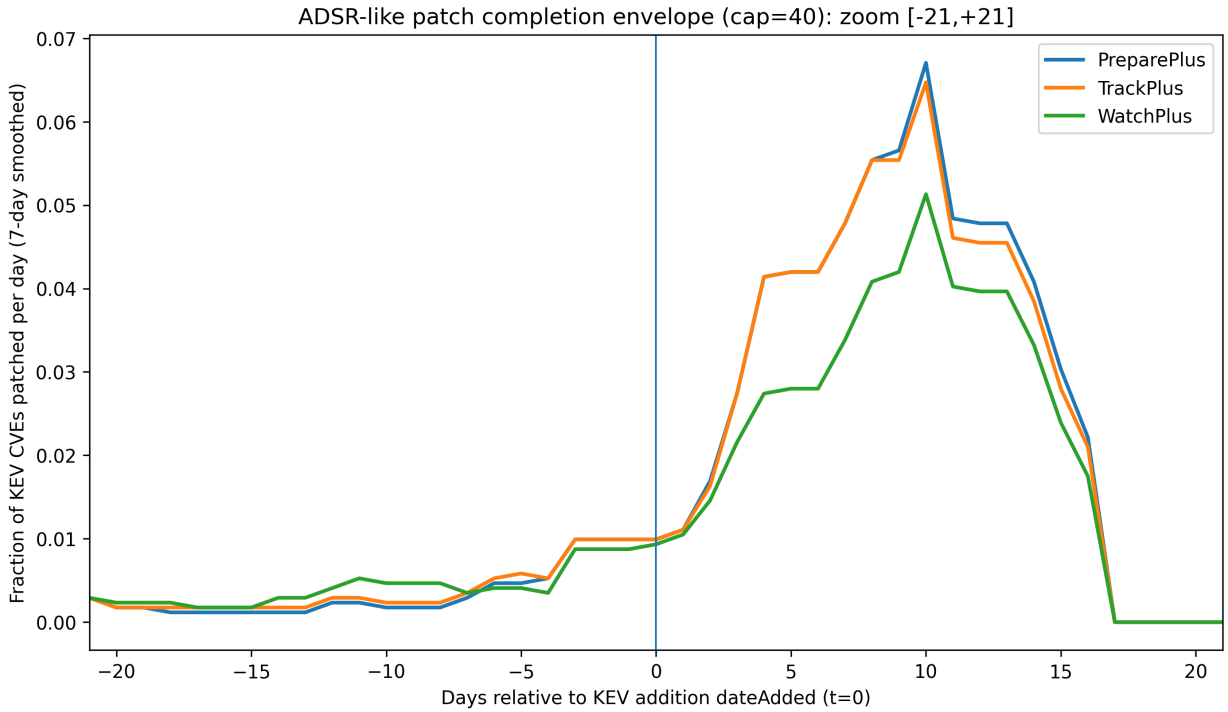

Figure S5: Completion density around KEV `dateAdded`. Kernel-smoothed completion density illustrates whether remediation concentrates at KEV or is distributed earlier by preparation.

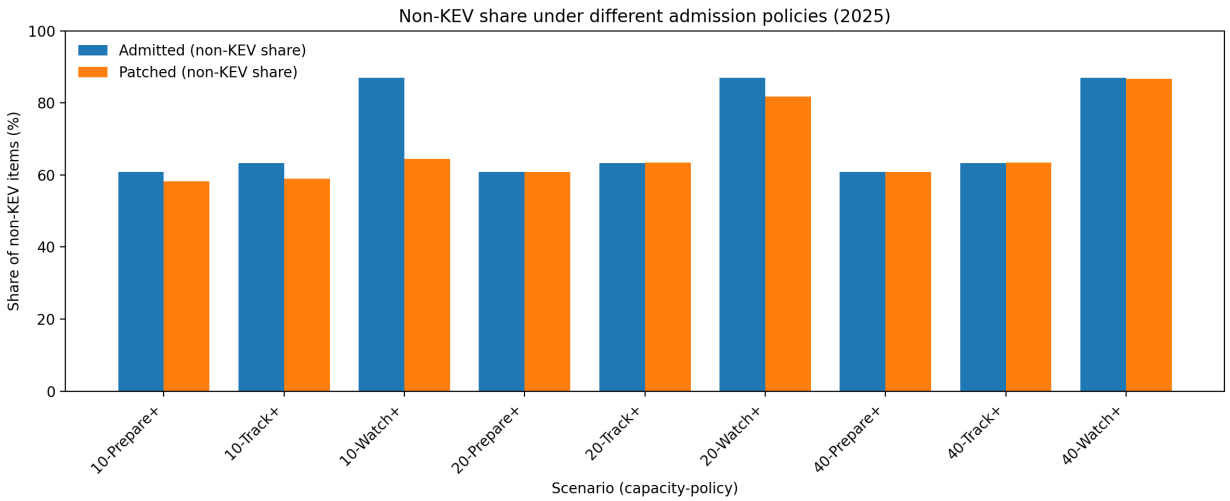

Figure S6: Non-KEV share of work (upper bound). Share of admitted and patched items that are not in KEV-2025, by scenario. Since KEV is incomplete, this should be interpreted as an upper bound on "overhead".

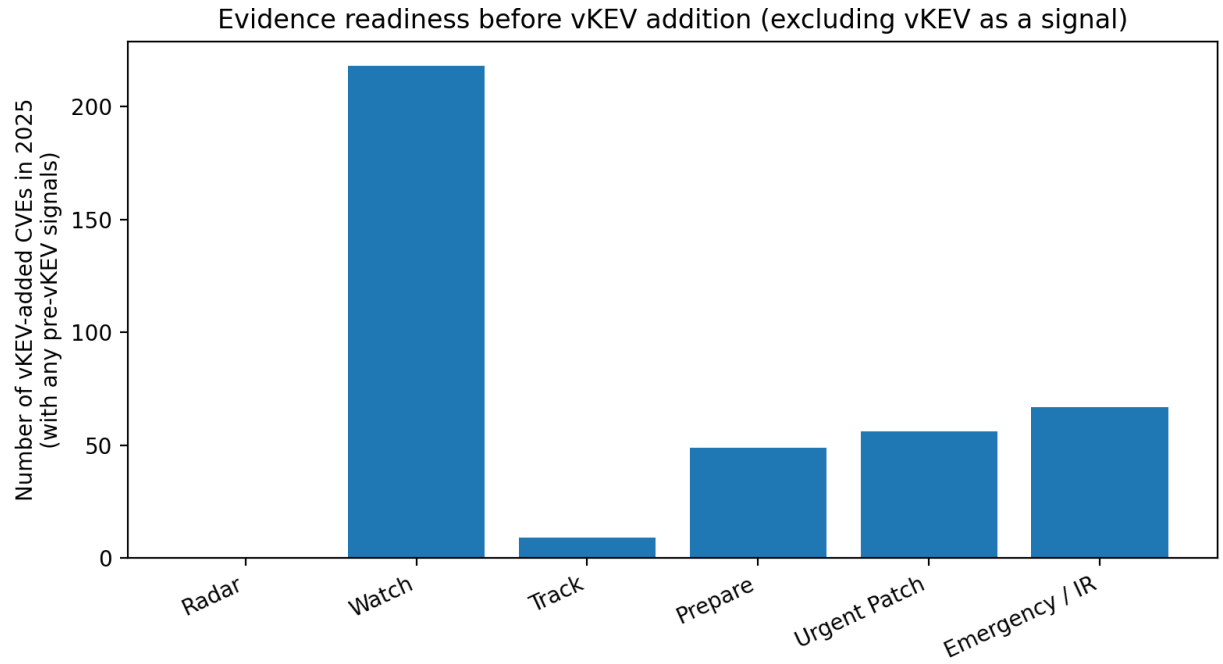

Figure S7: Pre-vKEV evidence readiness without "using the answer key". For vKEV additions in 2025 (N=888), we compute the UPS phase immediately before the vKEV listing timestamp while excluding the `VKEV_ADDED` event from the evidence stream.

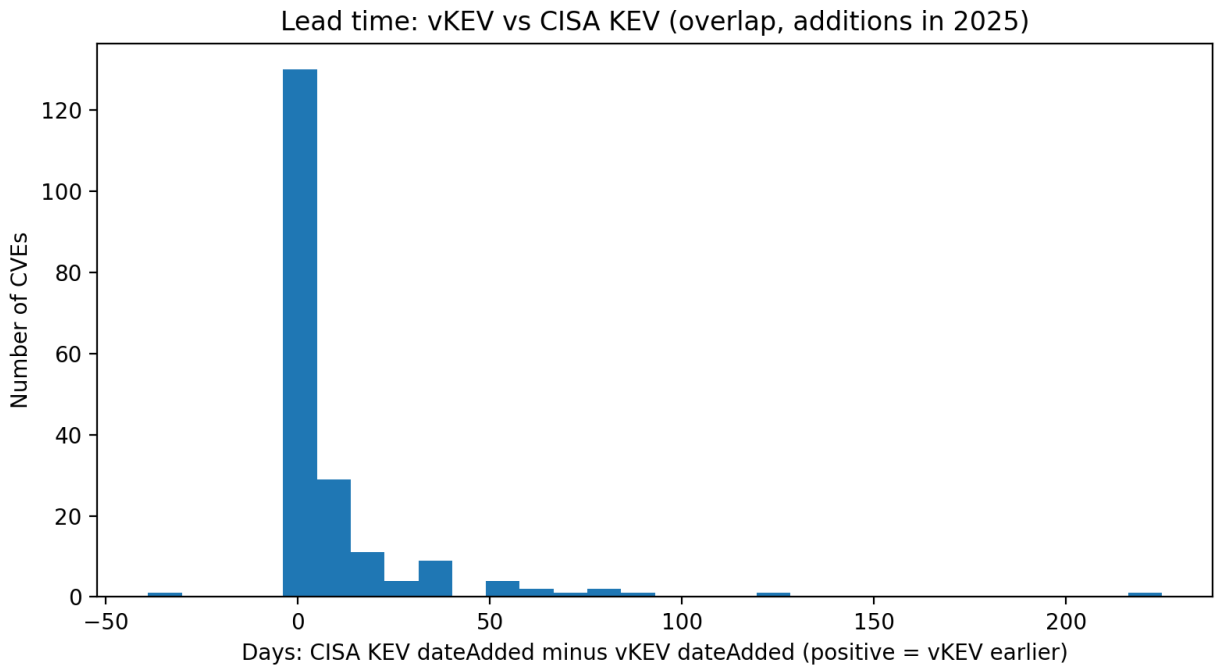

Figure S8: Lead time of alternative exploited catalogs. Distribution of (CISA KEV `dateAdded` − vKEV `dateAdded`) for CVEs appearing in both catalogs in 2025 (N=196). Positive values mean vKEV is earlier.

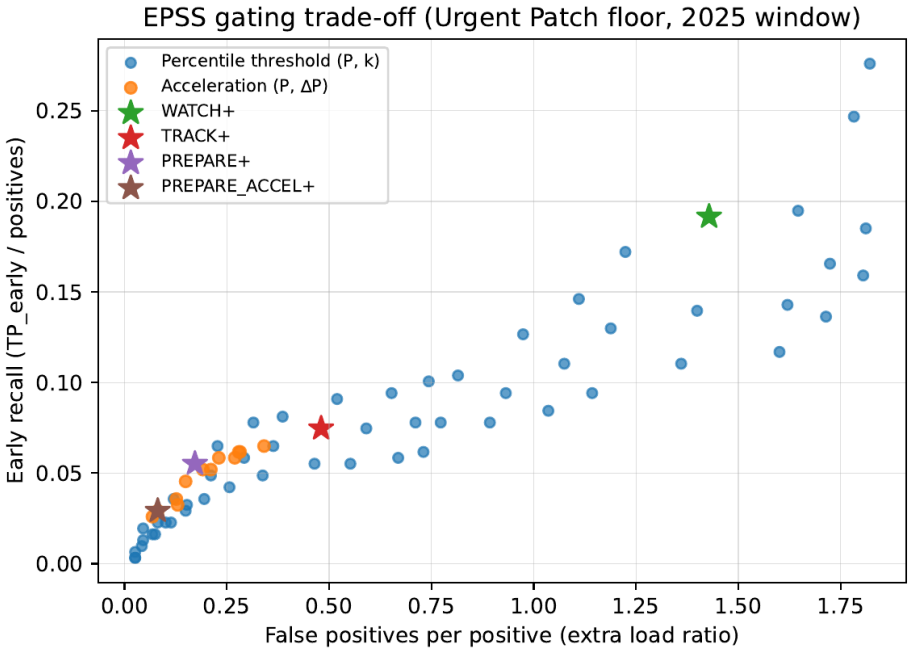

Figure S9: EPSS gating trade-off (Urgent Patch floor, 2025 window): early recall vs false positives per positive (extra load ratio), with selected operating points.

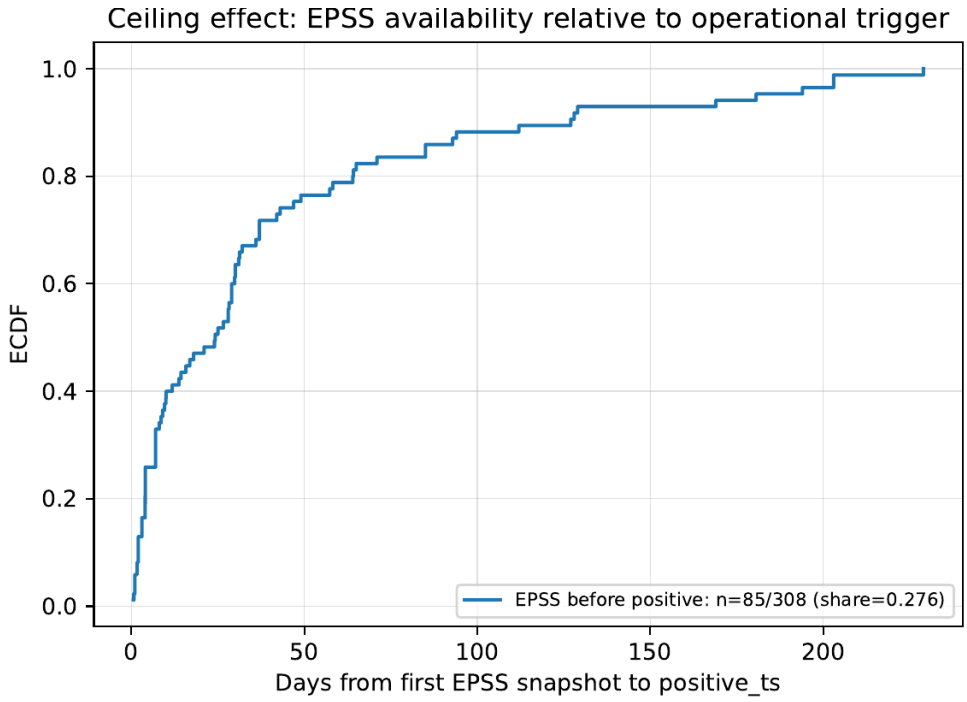

Figure S10: Ceiling effect: distribution of EPSS availability relative to the operational positive time (t_+).