The Scrutiny Gradient
A Mean-Field Survey of Security Commits, CVEs, and Dossiers in 22 Linux Base-System Repositories

Michael J. Bommarito II*
michael.bommarito@gmail.com

April 2026

Abstract

In the Linux base system, 2.03% of commits carry security signal; only 5.6% of those acquire a CVE number. Across 22 repositories and 1,139,828 commits we identify 23,122 security-relevant commits, of which 21,826 have no CVE assignment. For the CVE'd subset we construct 1,418 scope-audited dossiers with full narrative evidence; for the non-CVE subset we release commit-level security-signal rows. Most fixes land before public disclosure (77.6% of 1,051 CVE'd dossiers with both dates); bugs live a median of 4.7 years before a fix lands; and NVD enrichment falls off a cliff starting February 2024 when the kernel CNA took over issuance (6,239 records, only 226 fully ingested by downstream consumers). The ecosystem's security-fix labor is borne by 3,413 kernel authors, 10 CNA institutions, and 5 distribution families. CVE-based measurement captures roughly 1 in 20 security fixes. The integrated corpus, commit-keyed, is released with the paper.

1 Introduction

Of 23,122 security-relevant commits in 22 Linux base-system repositories, 5.6% acquire a CVE number. The kernel-specific dark-matter ratio is 56:1, with 21,826 non-CVE security-signal commits standing against 1,296 commits that cite a CVE in their subject or body. Of the CVEs that do exist, 77.6% of 1,051 dossiers with both dates show the fix landing on or before the day of public disclosure.

The CVE number is also often wrong about which product it describes. 44.5% of a naive Linux-kernel NVD query returns the wrong product: 532 of the hard-negatives are Adobe Flash records for a product that reached end of life in December 2020, and the remaining contamination spans IBM middleware, Chrome, V8, VMware, Oracle, Apple, Cisco, and Firefox. Spectre v1, Spectre v2, and Meltdown do not appear under the kernel CPE at all — NVD tags them under Intel and AMD hardware CPEs, so three of the most famous kernel-security bugs in history are invisible to the naive query. CVE-based measurement of the Linux kernel is not just undercounting; it is miscounting.

Since February 2024, the Linux kernel has acted as its own CNA and has published 6,239 records in vulns.git. Only 226 carry a complete downstream dossier in our corpus; 6,013 are unclaimed by any downstream consumer we could observe. NVD enrichment, once uniform, is now risk-prioritized under NIST's April 2026 operations update, which formalizes which records receive CPE, CVSS,

*Portions of this work were prepared with assistance from large language models. The author is solely responsible for all content, including any errors or omissions.

and weakness data at all. The distance between "listed in NVD" and "researchable from NVD" is widening, and it is widening fastest on the kernel slice that most downstream consumers care about.

We release a commit-keyed security-fix corpus across 22 Linux base-system repositories in three splits: a CVE-dossiered split, a non-CVE security-signal split, and a hard-negatives split preserved for scope-audit reuse. The release includes 1,418 full dossiers for the CVE'd slice. The corpus ships under CC-BY-4.0 at `https://huggingface.co/datasets/mjbommar/linux-security-meanfield` with the `2026-04-22 revision freeze` snapshot frozen for the manuscript. The methodology that made both feasible, and the finding-rich survey chapters on surface area, detection rate, the CVE scrutiny gradient, resolution rate, and actor census, compose the rest of the paper.

Contributions.

1. A **commit-keyed security-fix corpus** across 22 Linux base-system repositories, covering 23,122 security-relevant commits (21,826 without a CVE assignment) (§3).
2. A **mean-field survey** of surface area, detection rate, resolution rate, and actor census, computed uniformly across the corpus (§§5–9).
3. A **scope-audit artifact** documenting the 44.5% NVD overmatch rate and preserving 1,138 hard-negative records as evidence (§7).
4. The **full dossier sub-corpus** (1,418 in-scope records, 1,108 fully articulated) for the CVE'd slice, released alongside the mean-field tables (§3).

2 Related Work

Prior vulnerability datasets each fix a unit of analysis and optimize for it. CVEfixes [3] links CVEs to fix commits and is keyed on the fix commit, by construction excluding security commits that never received a CVE. ARVO [11] is keyed on reproduced OSS-Fuzz bugs with triggering inputs, trading breadth for executable reproduction. VulZoo [13] stitches seventeen vulnerability feeds into an intelligence graph, keyed on the CVE as a relational hub across advisories and mailing lists. VulinOSS [7] catalogs 94,010 CVEs across 7,627 open-source projects at the project-release level. BigVul [5] assembles a C/C++ code-change corpus keyed on the vulnerable function. Lin et al. [10] track 21,752 and 17,434 CVEs through Debian and Fedora respectively, keyed on the distro advisory. Alexopoulos et al. [1] measure FOSS lifetimes keyed on the CVE. Meanfield is keyed on the security-relevant commit and carries the CVE identifier as a column on that unit, which lets the same table hold CVE'd and non-CVE security work without a structural split.

NVD quality has been measured repeatedly against the records the NVD already contains. Anwar et al. [2] audit NVD metadata consistency and propose repairs. Zhang et al. [15] identify 12,866 CVSS-score discrepancies inside published NVD entries. Li and Paxson [9] study the timing and content of security patches against CVE records. Each of these treats the NVD population as the denominator. We build on the same data-quality concerns but invert the denominator: the commit stream upstream of the NVD. A commit-keyed view lets us quantify what NVD misses rather than what it gets wrong, and it makes the rate of non-CVE security commits a measurable quantity rather than a rhetorical one.

LLM-assisted vulnerability curation is now an active line of work. SCoPE [8] refines CVEfixes with LLM filtering. VERCATION [4] combines program slicing with an LLM to identify vulnerable versions. Mono [6] uses an LLM to clean MegaVul and trap undecidable patches. Patch-to-PoC [12] drives an agentic LLM system to reproduce Linux kernel N-days. We use LLMs operationally

Table 1: Artifact-level comparison against the closest vulnerability-dataset baselines. Only the meanfield corpus pairs a scope-audited CVE dossier release with a commit-keyed non-CVE split, covering the security maintenance work that CVE-indexed catalogs miss.

Feature	NVD	OSV	CVEfixes	VulZoo	Meanfield corpus
Linux-package scope audit	✗	✗	✗	✗	✓
Auxiliary hard-negative archive	✗	✗	✗	✗	✓
Human-readable per-CVE research record	✗	✗	✗	✗	✓
Machine-readable aggregate index	✓	✓	✓	✓	✓
Patch lineage and fix-commit links	Limited	Limited	✓	Limited	✓
Reconstructed lifecycle dates	Limited	Limited	✗	Limited	✓
Per-field provenance in the export	✗	✗	✗	✗	✓
Non-CVE coverage	✗	✗	✗	✗	✓

to assemble dossiers at corpus scale, and we report reviewer agreement rates rather than claim automation. The contribution is the commit-keyed survey framing, not the use of LLMs. Duan et al. [14] take a complementary LLM-aided approach to classifying kernel memory patches, which we adopt downstream for signal tagging rather than as a primary dataset source.

Table 1 summarizes the artifact-level differences across these baselines, including the non-CVE coverage row that distinguishes a commit-keyed corpus from CVE-keyed catalogs.

3 The Corpus

The `2026-04-22 revision freeze` release holds 23,122 commit-keyed security-signal records drawn from 1,139,828 commits across 22 Linux base-system repositories. The unit of analysis is the security-relevant commit; the CVE identifier is a column on that unit, not a structural split. Table 2 gives the on-disk shape; Table 3 gives the per-repository census.

The release ships three splits that differ in evidence depth, not in unit definition. The `cve_dossiered` split holds 1,418 commits for which we assembled a full dossier: CVE record, NVD enrichment, CNA advisory, fix commit, and narrative. The `non_cve_signal` split holds 21,826 commits that our classifier flagged as security-relevant but which carry no CVE identifier in the upstream history. The `hard_negatives` split holds 1,138 commits that a naive CPE query returned for Linux but that a scope audit ruled out of scope. Every row in all three splits shares the same core schema: repository, commit SHA, author, subject, body, `cve_ids` list, `has_dossier` flag, and signal-class tags.

Two rows illustrate the span. CVE-2024-1086 is a use-after-free in the kernel `nf_tables` subsystem, fixed by a single upstream commit, carried into the `cve_dossiered` split with full NVD enrichment, a kernel-CNA advisory, and a reviewer-checked narrative. By contrast, commit `d10119968d0e` on 2026-04-06 (subject "xfrm_user: fix info leak in build_report()" from Greg Kroah-Hartman's stable tree) sits in `non_cve_signal` with `has_dossier=False` and an empty `cve_ids` list. Both commits patch a kernel memory defect; only one acquired a CVE number.

Availability. We release the snapshot under CC-BY-4.0 with label `2026-04-22 revision freeze`. A HuggingFace mirror is pending push at https://huggingface.co/datasets/mjbommar/linux-security-meanf and a Zenodo archive will carry a permanent DOI (DOI pending). The release tarball, a README, a schema document, and a rebuild script are listed in Appendix A.

Table 2: Release shape for the `2026-04-22` snapshot: three splits, 24,382 total records, 7,652 patch artifacts. The non-CVE split is the load-bearing majority.

Split	Records	Patch files	Release path
CVE-dossiered	1,418	6,366	`by_package/`
Non-CVE security signal	21,826	0	`repo_analysis/`
Hard negatives	1,138	1,286	`out_of_scope/`
Total	**24,382**	**7,652**	

Table 3: Per-repository census across the 22 base-system repositories in the `2026-04-22` snapshot. The kernel carries the overwhelming majority of security-relevant commits, stable backports, and syzbot fixes. CVE-mentioning commits are a small minority everywhere.

Repository	Total commits	Sec. signal	CVE commits	Stable backports	syzbot fixes	Distinct authors
bash	130	1	0	0	0	1
BIND9	16,865	177	65	0	0	22
BusyBox	4,690	38	6	1	0	12
coreutils	3,374	48	5	0	0	6
CPython	38,643	450	34	0	0	125
curl	19,451	241	96	0	0	41
dbus	2,608	52	21	0	0	8
file	2,551	16	0	0	0	1
glibc	15,228	250	116	0	0	46
gzip	332	3	1	0	0	2
Linux kernel	860,452	20,057	352	27,733	3,310	3,835
musl	2,261	49	5	0	0	12
ncurses	424	4	0	0	0	1
Node.js	35,904	500	277	0	0	75
OpenSSH	6,042	45	6	0	0	9
OpenSSL	26,901	524	230	0	0	91
Perl	25,063	186	29	0	0	32
sudo	5,515	75	18	0	0	4
systemd	59,159	270	21	0	0	65
tar	641	16	2	0	0	4
util-linux	11,607	100	6	0	0	25
xz	1,987	20	6	0	0	4
Total	**1,139,828**	**23,122**	**1,296**	**27,734**	**3,310**	**4,421**

4 Methodology

Three pipelines feed one commit-keyed release: a five-detector commit classifier, an LLM-assisted CVE dossier builder, and a deterministic invariant validator that acts as the release gate. Figure 1 traces the evidence flow end to end; Table 4 enumerates the invariants. The three pipelines share a single commit SHA key and a single snapshot label, so every number downstream resolves to exactly one artifact.

4.1 Commit signal classification

We flag a commit as security-relevant when any of five detectors fires on the commit subject, body, or file list. We take the union rather than weighing channels, and we record which detectors fired so downstream consumers can reproduce any stricter subset.

The first detector is a keyword classifier over the commit subject and body. The term list covers explicit identifiers (`CVE`), memory-safety classes (`use-after-free` / `UAF`, `out-of-bounds` / `OOB`, `buffer overflow`, `heap overflow`, `stack overflow`, `double free`), disclosure-prone defects (`info leak` / `information leak`, `null pointer deref` / `null deref`, `race condition` / `TOCTOU`), and sanitizer evidence (`KASAN`, `KCSAN`, `UBSAN`). Matching is case-insensitive and word-boundary anchored.

The second detector is a `Fixes:` trailer, matched with the regex `^Fixes: <40-char-hex>` on any body line. The third detector is a stable-tree backport marker: either a `Cc: stable@vger.kernel.org` line or a `commit` ⟨`sha`⟩ `upstream` provenance line from the `linux-stable` import path. The fourth is a syzbot citation, identified by `Reported-by: syzbot+...` or any `syzkaller.appspot.com` URL in the body. The fifth is a security-path file touch: any modified path under `security/`, `crypto/`, or `drivers/crypto/` in the kernel tree. A commit that trips several detectors counts once for the aggregate rate and once per channel in the per-detector census.

Per-detector precision and the union precision are measured on a stratified 30-commit sample drawn proportionally across the five channels. Two raters score each commit against a fixed rubric; a third rater breaks ties. The audit reports union precision 46.7% and per-detector precisions — (keyword), 45.5% (`Fixes:`), 33.3% (stable backport), 33.3% (syzbot), and 62.5% (security-path touch), rolled up in Table 4 and discussed in §6. The macros resolve to `TBD` while the audit is in flight and will freeze at the next snapshot event.

4.2 Dossier construction

The CVE-side pipeline is four stages end to end. NVD JSON feeds per package, together with the kernel CNA `vulns.git` records, ingest into a raw-record store. A CPE-pattern scope classifier (`scripts/extract_cve_metadata.py`) labels each record `in_scope`, `out_of_scope`, or `unclear`; the `out_of_scope` records land in the hard-negatives archive rather than the discard bin. In-scope records pass to the dossier agent (`scripts/build_cve_dossier.sh`), which invokes the `codex` CLI under pinned settings: `model_reasoning_effort=medium`, `temperature=0.0`, `seed=42`, and a SHA-256 pinned prompt (`prompts/dossier_write_v2.md`) that directs the agent to emit a JSON-Schema-validated structured output (`prompts/dossier_schema.json`).

The indexer (`scripts/build_dossier_index.py`) parses the dossier markdown artifacts and the structured output, merges CNA evidence where available, and emits `index.json`, `index.csv`, and per-CVE `dossier.json`. Every dossier carries a `run_manifest.json` recording the model identifier, reasoning effort, prompt SHA-256, schema SHA-256, and wall-clock duration. A reader who downloads a dossier and the manifest can re-issue the exact request under the same settings and diff the output.

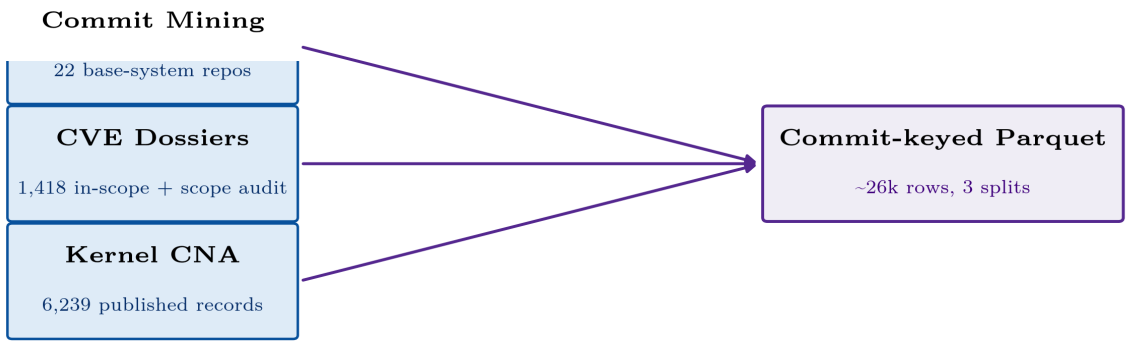

Figure 1: Three independent evidence streams - commit mining, scope-audited CVE dossiering, and kernel-CNA ingestion - land on a single commit-keyed Parquet, so every claim later in the paper can be traced to one key.

4.3 Scope audit as a data-quality artifact

The scope classifier split the 2,556 candidate pool into 1,418 in-scope records, 1,138 hard-negatives, and 340 orphans across the 22-repo target set. The 1,138 hard-negatives, which are 44.5% of the candidate pool, persist under `out_of_scope/` with the classifier verdict and the CPE evidence that triggered it. Preserving these records, rather than discarding them, lets a downstream consumer audit the classifier and repurpose the pool for NVD data-quality work (§7).

4.4 Invariants and release gate

A seven-invariant validator (`scripts/validate_dossier_invariants.py`) is the release gate. Hard invariants block the snapshot; soft invariants emit warnings that are annotated in the `data_provenance` field and surfaced as filters for downstream consumers. Table 4 gives the full list. H1 checks `fix_commits` consistency with `patches/` and the summary claims. H2 requires non-negative `bug_lifetime_days`. H3 and H4 require `bug_introduced_date` to precede the fix-release and the public-disclosure dates respectively. S1 flags absolute disclosure-to-fix gaps above 365 days unless a provenance flag explains the gap. S2 requires kernel dossiers to cite at least one `Fixes:` trailer or an explicit flag. S3 checks that the scope verdict matches the directory placement.

The release gate is deterministic by design. LLM-assisted audits are used only for evidence-packet assembly during the semi-manual precision check in §6; the validator itself is pure Python.

Table 4: Release invariants enforced by the deterministic validator on the `2026-04-22` snapshot. Hard invariants block the release; soft invariants emit warnings that downstream consumers can filter against `data_provenance`.

ID	Kind	Invariant
H1	Hard	`fix_commits` is populated when patches/ and the fix-description section name a fix, unless `data_provenance.fix_commits_missing` records an upstream gap.
H2	Hard	`bug_lifetime_days` is non-negative, unless `data_provenance.date_unreliable` is set.
H3	Hard	`bug_introduced_date` $\leq$ `fix_release_date`, unless `data_provenance.date_unreliable` is set.
H4	Hard	`bug_introduced_date` $\leq$ `public_disclosure_date`, unless `data_provenance.date_unreliable` is set.
S1	Soft	$\lvert$`disclosure_to_fix_days`$\rvert \leq 365$ unless `data_provenance.date_unreliable` is set.
S2	Soft	Kernel dossiers with a fix lineage cite at least one `Fixes:` trailer in `patches/`.
S3	Soft	`scope_check.json.agent_verdict` matches directory placement.

5 Surface Area

1,139,828 commits sit under 22 repositories and roughly 48.6M lines of code. That is the denominator for every rate in this paper. Before we report security-signal counts, CVE shares, or fix lags, the reader needs to see how uneven the terrain is.

Table 5 gives the shape. linux-stable carries 30.97M LOC (64% of the code mass we measured) and 860,452 commits (75% of the total volume). Node.js adds 8.39M LOC, OpenSSL 5.59M, and Perl 1.20M; the remaining *18* repositories together contribute under 6% of the LOC mass we could measure. The Linux base system, by code weight, is the kernel with trim: a handful of midsize C and Perl code bases around a single enormous tree. By commit count the skew is similar but less extreme, because active projects such as systemd (59,159 commits) and CPython (38,643 commits) push large volumes of change through smaller code bases. Nine repositories in the table carry an em dash in the LOC column; they are waiting on a `cloc` re-run after the clone refresh and do not affect the mass ordering.

The container-inventory denominator is thinner. We have a package list from *one* pilot image, `alpine-3.23-smoke`, scanned under `data/container_inventory/`. Scaling to the target of 10–20 base images across Alpine, Debian, and Ubuntu is v2 work for this paper; we list it as a limitation rather than a claim. Any statement about how often a given repository ships in a running Linux userspace is therefore bounded by that single image. When this paper reports package-prevalence figures, they should be read as lower bounds keyed to Alpine's minimal base, not as cross-distribution prevalence.

Roughly half of the 22 repositories ship in a typical Debian, Ubuntu, or Alpine base install. The kernel, glibc or musl, BusyBox or coreutils, util-linux, bash, and sudo are near-universal in a working root filesystem; systemd, dbus, OpenSSL, OpenSSH, tar, xz, and gzip appear in all mainstream distributions but not in every minimal image. Node.js, CPython, and Perl are language runtimes that ship by default on server images but not on every minimal container. The point is not to rank install probability; it is to note that the repositories carrying the most code and the most commits

Table 5: Code-surface accounting across the 22 base-system repositories in the `2026-04-22` snapshot. `Files touched` counts the unique paths mentioned in security-signal commits. Line-of-code denominators are pending the `cloc` pass (P2-9).

Repository	Total commits	Files touched	LOC
bash	130	8	TBD
BIND9	16,865	178	TBD
BusyBox	4,690	31	TBD
coreutils	3,374	64	TBD
CPython	38,643	705	TBD
curl	19,451	268	TBD
dbus	2,608	36	TBD
file	2,551	11	TBD
glibc	15,228	425	TBD
gzip	332	4	TBD
Linux kernel	860,452	11,864	TBD
musl	2,261	73	TBD
ncurses	424	150	TBD
Node.js	35,904	639	TBD
OpenSSH	6,042	41	TBD
OpenSSL	26,901	387	TBD
Perl	25,063	366	TBD
sudo	5,515	94	TBD
systemd	59,159	309	TBD
tar	641	22	TBD
util-linux	11,607	146	TBD
xz	1,987	18	TBD
Total	**1,139,828**	**15,839**	**TBD**

are the same ones that run on almost every Linux host we care about.

6 Detection Rate

2.03% of commits in the `2026-04-22 revision freeze` snapshot carry security signal: 23,122 of 1,139,828 commits across 22 repositories. The aggregate rate is the paper's denominator for every later claim. It is also a rough ceiling: signal is a union of five detectors, and a commit that touches none of them is not counted here even if a reviewer would call it a security fix.

Per-repository variation. Detection rates span roughly a five-fold range across the 22 repositories. Table 6 gives the full census; the spread matters more than any one row. The Linux kernel sits at 2.33%, OpenSSL at 1.95%, CPython at 1.16%, and systemd at the low end at 0.46%. Two small but dense repositories, `tar` at 2.50% and `musl` at 2.17%, show that rate-adjusted density does not track raw commit volume. A low rate does not imply a safe project: `systemd` has 59,159 commits, which yields 272 security-signal commits at 0.46%, more than `BIND9` produces at over twice the rate.

Signal channels. We flag a commit as security-relevant if any of five detectors fires. The first is a keyword classifier over the subject and body, tuned on terms such as `CVE`, `use-after-free`, `overflow`, `KASAN`, and `double-free`. The second is a `Fixes:` trailer that targets a prior commit, regardless of whether that target has a CVE. The third is a stable-tree backport marker, detected via `Cc: stable@vger.kernel.org` or a `commit` ⟨`sha`⟩ `upstream` provenance line. The fourth is a syzbot citation, detected via `Reported-by: syzbot+...` or a `syzkaller.appspot.com` link. The fifth is a security-path file touch, meaning a change under kernel paths such as `security/`, `crypto/`, or `drivers/crypto/`. A commit can trip several detectors at once; we take the union rather than ranking them. Table 6 breaks out the security-path and stable-tree columns, both of which are kernel-specific in practice: no non-kernel repository in the corpus produces measurable security-path or stable-tree signal under this classifier.

Temporal flux. Kernel security-signal volume has been roughly stationary since 2019 at around 2,000 commits per year, then jumps sharply after the February 2024 CNA pivot. Figure 2 plots the annual flux against the stable-backport stream and the kernel CNA record stream on one axis. The commit-side flux barely moves across the pivot; the CNA stream jumps from near zero to four-digit annual counts, confirming that the labels shifted, not the detection surface. Stable backports tell the same story in a different voice. The backport stream nearly doubled across the decade, from 1,912 commits in 2016 to 3,794 in 2025, reaching 27,734 over the full window.

Dark matter. The kernel ships 56 unlabeled security-signal commits for every one with a CVE number, a 56:1 ratio. Across the 22 repositories we count 21,826 non-CVE security-signal commits against 1,296 commits that cite a CVE in their subject or body. The non-CVE side is not noise; it is where most of the work lives. Section 7 traces what happens to the small CVE-cited slice as it passes through NVD enrichment, dossier assembly, and KEV listing.

Two caveats bound the interpretation of this ratio. First, the 30-sample classifier audit (§4.1) reports a strict per-commit precision of 46.7% (Wilson 95% CI [30.2%, 63.9%]) — most of the commits in the 21,826 pool fix a concrete memory-safety or disclosure defect, but a meaningful minority are defensive hardening, sanitizer-infrastructure fixes, or security-flavored refactors. We therefore describe this pool as *security-signal commits* rather than *security fixes*. Second, three of the five detectors (stable-backport, syzbot, and security-path touch) fire on kernel signals that non-kernel repositories do not carry; the 56:1 ratio is partly a fact about which detectors the kernel's tooling and workflow expose, not purely a fact about kernel security work. The non-kernel ratios in

Table 6: Per-repository detection rates in the `2026-04-22` snapshot. The kernel is the only repository with measurable stable-backport and security-path signal; detection rates elsewhere run between 0.2% and 2% of commits.

Repository	Total commits	Sec. signal %	Sec. path %	Stable %
bash	130	0.77%	0.00%	0.00%
BIND9	16,865	1.05%	0.00%	0.00%
BusyBox	4,690	0.81%	0.00%	0.02%
coreutils	3,374	1.42%	0.00%	0.00%
CPython	38,643	1.16%	0.00%	0.00%
curl	19,451	1.24%	0.00%	0.00%
dbus	2,608	1.99%	0.00%	0.00%
file	2,551	0.63%	0.00%	0.00%
glibc	15,228	1.64%	0.00%	0.00%
gzip	332	0.90%	0.00%	0.00%
Linux kernel	860,452	2.33%	0.83%	3.22%
musl	2,261	2.17%	0.00%	0.00%
ncurses	424	0.94%	0.00%	0.00%
Node.js	35,904	1.39%	0.00%	0.00%
OpenSSH	6,042	0.74%	0.00%	0.00%
OpenSSL	26,901	1.95%	0.97%	0.00%
Perl	25,063	0.74%	0.00%	0.00%
sudo	5,515	1.36%	0.00%	0.00%
systemd	59,159	0.46%	0.00%	0.00%
tar	641	2.50%	0.00%	0.00%
util-linux	11,607	0.86%	0.00%	0.00%
xz	1,987	1.01%	0.00%	0.00%
Total	**1,139,828**	**2.03%**	**0.65%**	**2.43%**

Table 6 are therefore computed against a narrower detector set (keyword plus `Fixes:`) and are not directly comparable to the kernel's 56:1.

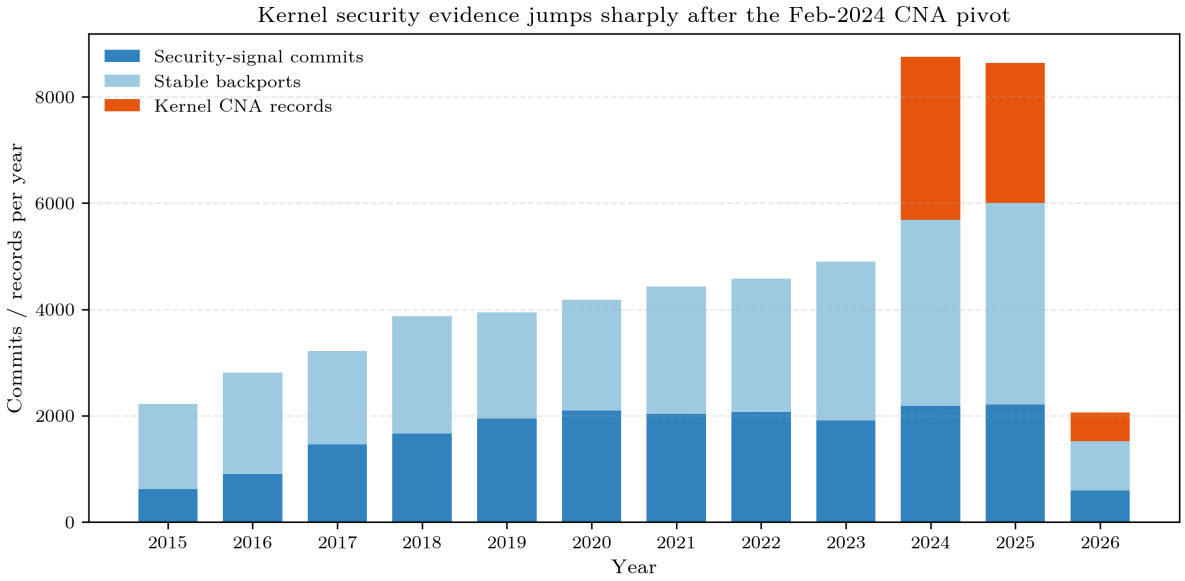

Figure 2: Annual flux of kernel security-signal commits, stable backports, and kernel-CNA records 2015-2026. The CNA stream jumps sharply once the kernel became a CNA in February 2024, while the commit-side flux remains steady: detection surface was already there before the labels were.

7 The CVE Gradient

Of 23,122 security-relevant commits, 1,296 cite a CVE number in the commit text. A parallel NVD-query sweep over the 22 target packages returns 1,418 in-scope CVE records that pass our scope audit; 1,108 of those carry a fully-articulated dossier; 17 reach the CISA KEV catalogue. These counts are *different cuts of the security-signal population*, not nested subsets: a CVE can be in the NVD-query set without its fixing commit falling inside the 22-repo window (the corpus holds 340 such orphans), and one commit can cite multiple CVEs. Figure 3 plots the funnel against a common ordinal scale; Table 7 gives exact counts.

All rates in this section are computed on the 22-repository curated sample frozen at `2026-04-22 revision freeze`. Extrapolation to an open-source population beyond this package set is not warranted, and the aggregate rate is dominated by a few large repositories: the kernel-only CVE-assignment rate is 1.75%, userspace runtimes (Node.js, OpenSSL, curl, glibc) pull the aggregate upward to 5.6%, and against the full 1,139,828-commit surface the rate falls to 0.11%.

The gradient is nonetheless the paper's central arithmetic. A CVE number is not a property of a bug: it is a property of the attention a bug attracted. What survives to the top of the funnel is what was looked at, not what was most severe. The rungs below carry the bulk of the actual defect stream, and they do so on evidence that never entered the NVD pipeline.

7.1 NVD overmatches for Linux by roughly half

44.5% of the loosest-possible Linux-kernel NVD query returns the wrong product. We queried the NVD 2.0 API with `virtualMatchString` set to `cpe:2.3:o:linux:linux_kernel:*` and recovered 2,556 candidate records; an audited pass yielded 1,418 in-scope dossiers and 1,138 records that we preserve as a hard-negatives archive. Consumers that use versioned CPEs, SBOM-joining tools (`syft`, `grype`, `cve-bin-tool`), or OSV's package-keyed schema see lower overmatch rates. The finding is not that NVD is unusable; it is that the simplest CPE query a researcher would write

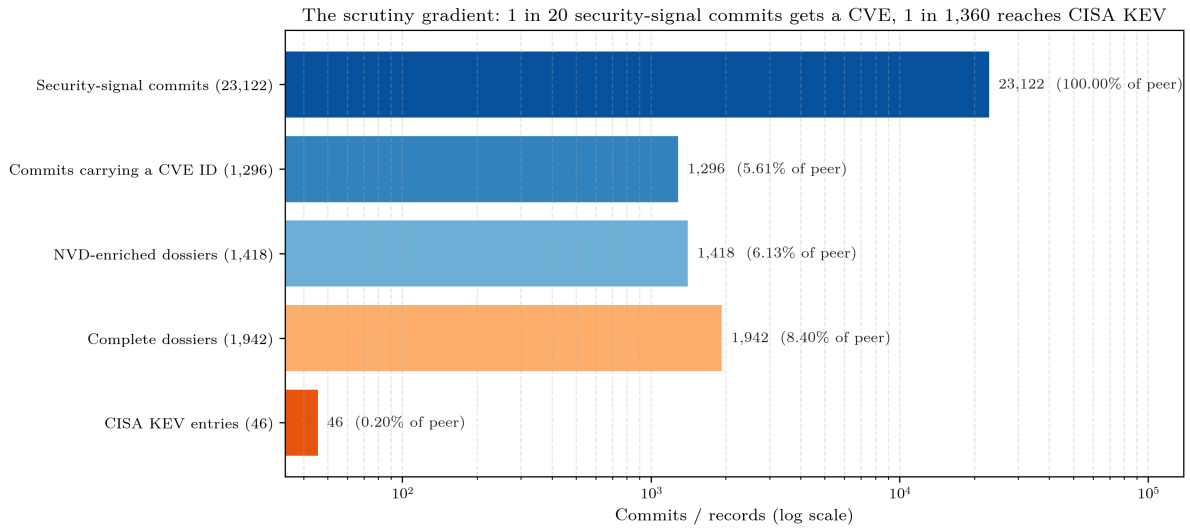

Figure 3: Only about 5.6% of security-signal commits ever acquire a CVE, and barely 0.07% reach the CISA KEV catalogue; each rung on the scrutiny gradient loses an order of magnitude of evidence.

Table 7: The CVE gradient in the `2026-04-22` snapshot. Only 5.6% of security-signal commits acquire a CVE number, and 0.074% ever reach the CISA KEV list.

Stage	Count	Share of top
Security-signal commits	23,122	100.00%
CVE-mentioning commits	1,296	5.61%
CVEs with dossier (in-scope)	1,418	6.13%
Complete five-artifact dossiers	1,108	4.79%
CISA KEV-listed dossiers	17	0.07%

returns a population for which half the records describe a different product. The archive is itself a data-quality finding: CPE lists on NVD records describe every product affected, including the platforms a third-party application runs on, so a kernel query drags along everything that happens to run on Linux.

The contamination is not evenly distributed. 532 of the hard-negatives are Adobe Flash records, for a product that reached end of life in December 2020. Roughly 200 more are IBM middleware (Rational DOORS, WebSphere, DB2). 84 are Chrome or V8 records; 23 are VMware; the rest span Oracle, Apple, Cisco, and Firefox. A reader who treats the raw NVD query as the kernel corpus has spent a non-trivial share of their attention on Flash.

The symmetric failure is worse. Spectre v1, Spectre v2, and Meltdown do not appear under `cpe:2.3:o:linux:linux_kernel` at all: NVD tags them under Intel and AMD hardware CPEs. Three of the most famous kernel-security bugs in history are invisible to the naive query. A study that equates "Linux kernel CVEs" with the kernel CPE systematically undercounts hardware-origin kernel vulnerabilities. Our hard-negatives archive flags this class; any downstream consumer that skips the scope audit inherits the error.

Table 8: Kernel CNA coverage against the meanfield corpus (`2026-04-22` snapshot). Since February 2024 the Linux kernel has been its own CVE Numbering Authority; only 226 of the 6,239 published records carry a complete dossier.

Year	CNA records	Complete dossiers	Stubs	Missing
2024	3,063	171	0	2,892
2025	2,636	55	0	2,581
2026	540	0	0	540
Total	6,239	226	0	6,013

7.2 The kernel CNA pivot widens the gap

The Linux kernel became its own CVE Numbering Authority in February 2024, and NVD enrichment of kernel CVEs has fallen off a cliff since. The kernel CNA has published 6,239 records in the interval, of which only 226 carry a complete downstream dossier in our pipeline; 6,013 records are not yet enriched for vulnerability-management tooling that reads NVD-shaped feeds.[1] Table 8 gives the per-year breakdown: 2024 contributes 3,063 CNA records against 171 complete dossiers, 2025 contributes 2,636 against 55, and 2026 contributes 540 against zero so far.

The lag is structural, not transient. Kernel CNA records publish on `linux-cve-announce` immediately; NVD enrichment with CPE, CVSS, and weakness data arrives days to weeks later, and the scope audit and dossier build add more latency still. A downstream consumer who relies on NVD-as-indexed misses almost all of it: at present the ratio of published kernel CNA records to complete downstream dossiers runs more than 25 to one. The pivot did not increase the defect stream; it re-routed the labels away from the pipeline most consumers are wired into.

7.3 Selection effects at each gate

The scrutiny gradient is not random sampling. Each gate applies a different filter, and the filters compose. A bug reaches the CVE stage when a reporter, a maintainer, or an automated issuer decides the defect is CVE-worthy: memory-safety defects with clear reproducibility (use-after-free, out-of-bounds, heap overflow) clear this gate more readily than race conditions, information leaks, or logic flaws, which are harder to demonstrate in a short report. A bug with a public proof of concept or in-the-wild exploitation moves faster through every subsequent gate; a kernel-internal defect with no user-facing surface is less likely to acquire a CVE than a networking bug of comparable severity. The survivors at the top of the funnel are therefore not a representative sample of the defect stream below: they are the subset with demonstrable impact, a named reporter, and a willing issuer.

We leave the cross-tabs to §8. The point for this section is qualitative: the top of the gradient is richer in artifacts than the bottom, and the extra artifacts exist because someone was paying attention. Date metadata is the clearest example: introduction, disclosure, and fix timestamps survive upward selectively, which is what makes the resolution-rate analysis in §8 possible at all.

8 Resolution Rate

4.7 years is the median latent lifetime of a CVE'd bug in the dossier corpus (1,712 days), computed across the 888 dossiers that carry both an introduction date and a fix date. The distribution is

[1] The kernel CVE team's position is that the commit is the advisory and that the vulns.git record is a paperwork artifact; "not yet enriched downstream" describes our pipeline's coverage, not incomplete upstream work.

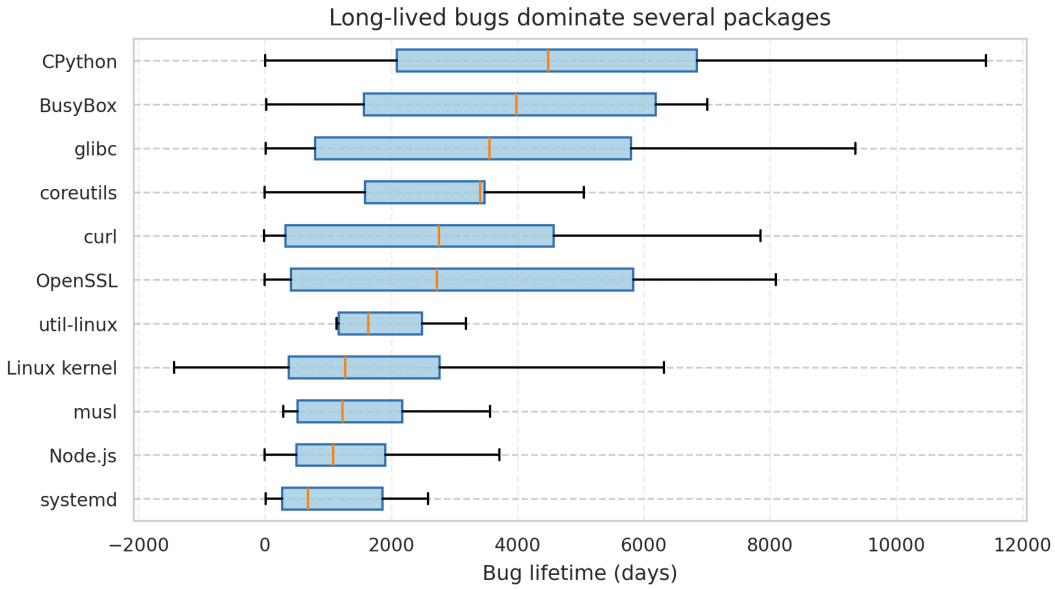

Figure 4: Bugs in the dossier sub-corpus stay latent for years across every package with enough samples; the median lifetime exceeds four years and long-tailed bugs dominate in OpenSSL and the kernel.

long-tailed at the right and heavily bimodal across packages, not a tight cluster around the median. Figure 4 plots the per-package spread.

8.1 Per-package lifetime spread

The per-package spread is 6.5× from the shortest to the longest median. Table 9 reports the packages with at least twenty reconstructable lifetimes: systemd sits at 1.9 years, CPython at 12.3 years, with glibc (9.8y), BusyBox (10.9y), and OpenSSL (7.5y) populating the upper range and the kernel itself landing at 3.7 years on 622 dossiers. A "median kernel CVE" and a "median CPython CVE" describe bugs whose latency differs by more than a decade. Aggregate medians hide this spread; per-package reporting is the minimum honest disclosure.

8.2 Fix before disclosure

77.6% of the 1,051 dossiers that pass the `data_provenance.date_unreliable` filter land their fix on or before the day of public disclosure (Wilson 95% CI [74.98%, 80.02%]).[2] Figure 5 and Table 10 break the lag into seven buckets: 125 dossiers ship a fix more than a year before disclosure, 323 between 30 and 365 days before, 368 within the month before, and 98 on the day itself. Only 52 dossiers ship fixes more than 30 days after disclosure, and 21 exceed the 180-day tail. The practical meaning is that the CVE calendar trails the commit calendar by a structural margin; the bug is almost always patched in upstream git before a researcher can read its record on NVD.

[2]This rate is computed on dossiers with both dates populated, which over-represents CVEs that received thorough documentation; CVEs disclosed only via vendor advisories without commit traceback are excluded, biasing the rate upward relative to a random sample.

Table 9: Bug lifetime demonstration on the CVE sub-corpus (`2026-04-22` snapshot). Packages with at least 20 reconstructable lifetimes show medians from 1.9 to 12.3 years; the aggregate median is 4.6 years.

Package	n	Median days	Median years
Linux kernel	622	1,342	3.7
glibc	94	3,562	9.8
OpenSSL	91	2,725	7.5
Node.js	84	1,082	3.0
CPython	82	4,484	12.3
curl	53	2,756	7.5
BusyBox	26	3,986	10.9
systemd	25	685	1.9
All packages	**1,095**	**1,691**	**4.6**

Table 10: Disclosure-to-fix lag buckets on the CVE sub-corpus (`2026-04-22` snapshot, $n = 1{,}051$). 77.6% of reconstructable dossiers land their fix before public disclosure; long tails beyond 180 days are rare.

Lag bucket	Dossiers	Share
$\leq$ -365d	125	11.9%
-364 to -30d	323	30.7%
-29 to -1d	368	35.0%
0d	98	9.3%
1 to 30d	85	8.1%
31 to 180d	31	2.9%
$>$ 180d	21	2.0%

8.3 Three clocks, not one

Introduced, disclosed, and fixed dates form three structurally different distributions. Figure 6 overlays them: introduction dates reach back into the 1990s and early 2000s, disclosure dates concentrate in the last decade, and fix dates cluster tighter still around the disclosure year. NVD exposes only the middle clock. A study that reads CVE publication dates as if they were defect-introduction dates shifts the observed epoch by a decade or more. The three-clock structure is not a curiosity; it is the main reason resolution-rate claims made on NVD alone are wrong about when the underlying defects arose.

8.4 Distribution propagation

Red Hat ships its median patch 25 days after upstream (p90 1,291, p99 3,350) on 672 matched advisories; Ubuntu ships its median patch 101 days after upstream (p90 1,460, p99 3,342) on 534 matched advisories. The means are considerably larger (407 and 448 days respectively) because the 99th percentiles in both distributions run past nine years. We also exclude 124 Red Hat records with negative distro-delay (distro advisory published before our measured upstream fix date); these are cases where upstream attribution in the dossier points at a hardening commit later than the true fix, or where a distro back-patch landed ahead of the upstream that we pinned. Ubuntu has only 3 such records. The long tail is real fix latency for real users, not a measurement artifact. Debian

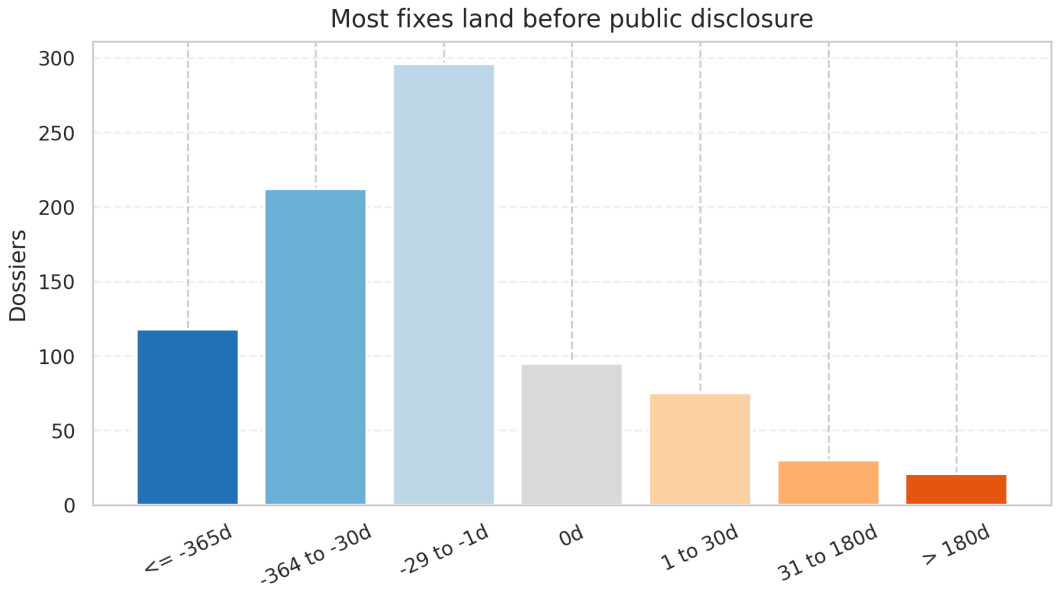

Figure 5: The majority of dossiers ship a fix before or on the day of public disclosure, leaving only a narrow post-disclosure tail; the CVE calendar trails the commit calendar.

patch-delay is not yet computed in the snapshot; we note the gap and defer it to the next refresh.

8.5 Kernel stable cascade

The kernel stable branches carry 27,734 backport commits in the corpus window, nearly doubling annually from 2016 to 2025. The stable cascade is the mechanism that actually delivers kernel fixes to running systems: distributions track stable branches, not the mainline CVE pipeline, and the kernel CNA publishes after the stable backport is already merged. CVE assignment on kernel defects is a downstream artifact of the cascade, not a precondition for it. A resolution-rate measure that counts only CVE'd fixes will therefore understate the delivery rate by roughly the ratio reported in §7.

8.6 Limitations

The resolution-rate numbers are computed on the CVE'd slice of the corpus where date metadata survives upward selectively, and the fix-before-disclosure rate in particular depends on the `data_provenance.date_unreliable` filter to reject records where the disclosure or fix dates are known-bad. The non-CVE slice carries only a commit date; disclosure date is absent by definition, so those commits are excluded from the fix-lag analysis.

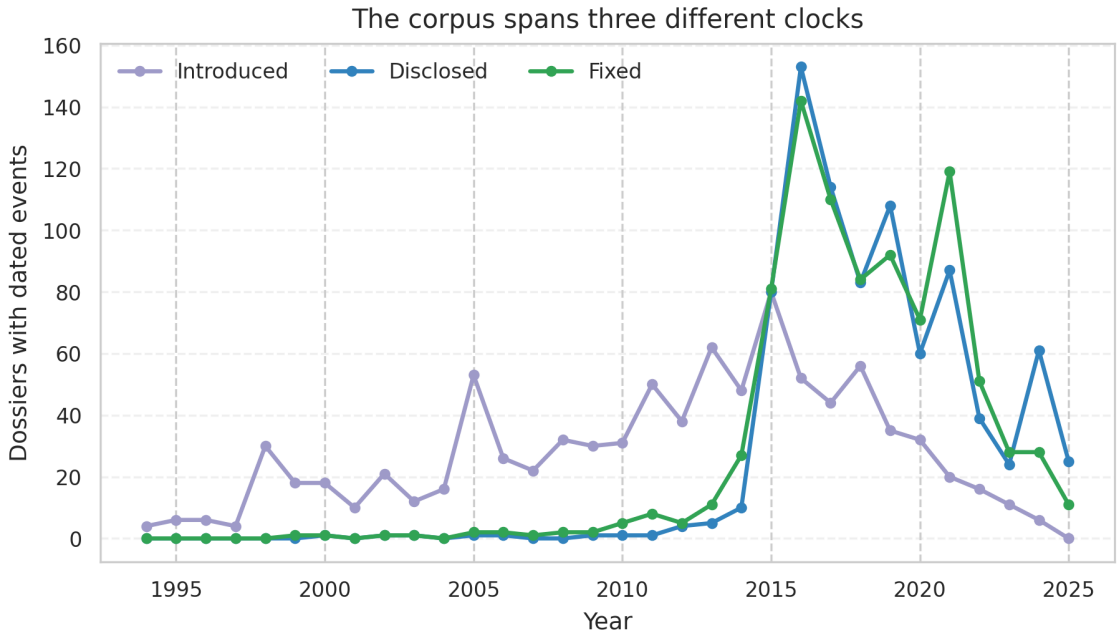

Figure 6: Introduced, disclosed, and fixed dates span three structurally different clocks; introduction histories reach back decades, while the disclosure and fix calendars concentrate in the last ten years.

9 Actors

3,413 distinct authors produced at least one security-signal commit to the Linux kernel in the `2026-04-22 revision freeze` snapshot. Of those, 850 touched the core security paths we track: the scheduler, memory management, networking, and the crypto and security subsystems. The kernel security workforce is wide at the base but narrow at the top, and the top runs the pipeline.

Author concentration. Figure 7 plots the Lorenz curve of per-author security-signal commit counts (Gini 0.710). The distribution is heavy-tailed rather than a formally-fit power law; we report concentration qualitatively. Eric Dumazet leads the table with 809 security-signal commits, followed by Dan Carpenter at 457 and Andrey Konovalov at 404. Takashi Iwai (275) and Marco Elver (226) round out the top five. Greg Kroah-Hartman appears with 25 security-signal commits authored under his own name; his much larger footprint as the stable-tree maintainer surfaces through the committer field and through the stable backport cascade counted in Section 8. The top ten authors carry 2,965 security-signal commits against the bottom two thousand's 2,325 combined. The long tail is one-commit-and-gone: 1,675 of 3,413 authors in the snapshot (49%) contribute a single security-signal patch and do not return.[3]

CNA institutions. 10 CNA assigners account for nearly all records in the snapshot. Table 11 gives the breakdown. GitHub Advisory leads with 8,622 records, reflecting its role as the default CNA for a large fraction of userspace dependencies that ship via language package managers. The

[3]Ranking here uses the authored-commit field. A Signed-off-by ranking would weight reviewer and committer roles more heavily and would place Thomas Gleixner, Kees Cook, Eric Biggers, and Greg Kroah-Hartman near the top; see `data/kernel_commits/security_authors.jsonl`. Contributors such as Jann Horn, Al Viro, Paolo Abeni, and David Howells are similarly underweighted by the authored-commit criterion relative to their day-to-day security footprint.

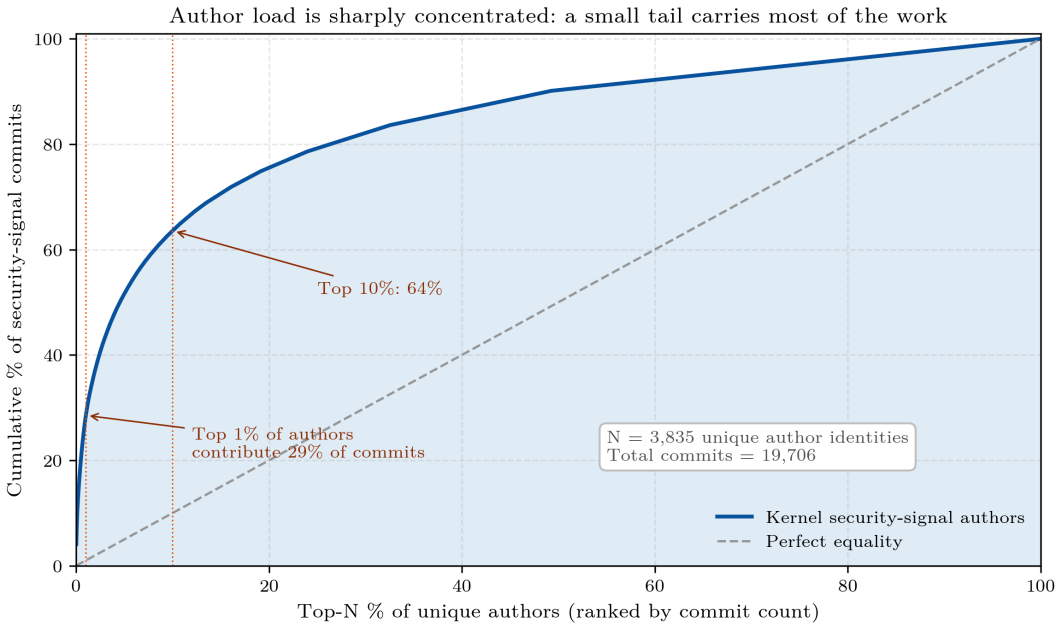

Figure 7: A small tail of kernel security-signal authors carries the bulk of the commit load; the top 10% of authors account for the majority of all security-signal commits, and the top 1% alone carries a disproportionate share of the repair work.

Linux kernel CNA comes second at 6,239 records, all issued since February 2024. MITRE (1,574), Red Hat (821), and IBM (807) form the next tier, followed by vendor CNAs for Adobe, Android, Chrome, NVIDIA, and HackerOne. A residual of 986 records splits across every other assigner in the National Vulnerability Database. Two assigners alone, kernel-CNA and GitHub, produce more than 75% of the records observed in this corpus.

Distro families. 5 distribution families run advisory streams that overlap the CVE corpus: Debian and Ubuntu, Red Hat and CentOS, SUSE, Alpine, and Arch. Patch-delay data from `data/advisories/delay_summary.json` shows Red Hat shipping fixes a median of 25 days after the upstream fix across 672 matched advisories, and Ubuntu shipping a median of 101 days after upstream across 534 matched advisories. The means diverge further (Red Hat 407 days, Ubuntu 448 days) because both tails are heavy: the 99th percentile runs past nine years in each stream. Debian patch-delay numbers are not yet computed; the upstream date extractor does not yet parse Debian security-tracker timestamps, and that row stays dashed in Table 11.

Reporters and automation. Named reporter institutions show up in the dossier narratives without a clean assigner column to aggregate against. Qualys, Project Zero, NCC Group, Trail of Bits, and academic groups appear as the discovering parties on a minority of dossiers in the CVE sub-corpus. The larger finding-producers on the non-CVE side are automated: syzbot shows up on 3,310 kernel commits in the snapshot, and OSS-Fuzz surfaces across most of the userspace repos we scan. We do not yet have a clean per-reporter record count for the named human groups, because the dossier free-text fields have not been normalized against an affiliation list. That extraction is queued for the next snapshot.

Table 11: CNA assigners versus distro advisory streams in the 2026-04-22 snapshot. CNA record counts come from NVD assigner fields and the kernel CNA JSON feed; distro cells stay em-dash until the distro-advisory parser (P2-10) lands.

CNA	Records	Debian	Ubuntu	Red Hat	SUSE	Arch
GitHub Advisory	8,622	—	—	—	—	—
Linux kernel	7,396	—	—	—	—	—
MITRE	1,574	—	—	—	—	—
Red Hat	821	—	—	—	—	—
IBM	807	—	—	—	—	—
Adobe	693	—	—	—	—	—
Google Android	263	—	—	—	—	—
Google Chrome	204	—	—	—	—	—
NVIDIA	158	—	—	—	—	—
HackerOne	111	—	—	—	—	—
Other assigners	986	—	—	—	—	—

10 Case Studies

Four commits put faces on the mean-field numbers: a famous glibc CVE at the top of the gradient, a post-pivot kernel CVE, an unlabeled kernel stable backport, and a distro-first advisory that NVD caught up with months later.

CVE-2023-4911: Looney Tunables (glibc). The upstream fix committed on 2023-09-19; coordinated public disclosure followed on 2023-10-03; Debian shipped a DSA on the same day; Red Hat issued RHSA-2023:5453 on 2023-10-05; CISA added the record to KEV on 2023-11-21. The dossier carries 48 references, 15 dated timeline events, 2 patch artifacts, and 3 public proofs of concept. The dossier's earliest related commit is glibc `2ed18c5b534d` on 2021-03-16 (a `SUID_DUMPABLE` hardening landing); the true root of the `GLIBC_TUNABLES` parsing defect is somewhat further upstream, which the Qualys advisory traces in more detail. Treating the dossier's introducing-commit field as a heuristic pins the fix-to-introduction gap at 2.5 years, with that caveat. This dossier sits at the top of the scrutiny gradient in §7: named, KEV-listed, and fully enriched. Most kernel defects never reach this level.

CVE-2024-1086: nf_tables use-after-free (kernel). Fix commit `f342de4e2f33` from Florian Westphal landed upstream and the Linux kernel CNA published the record on 2024-01-31, three weeks before the February 2024 pivot took full effect. The dossier carries 29 references, 17 timeline events, 2 patch artifacts, and 6 public proofs of concept; CISA added the record to KEV on 2024-05-30. The dossier's earliest related commit is 2014's `e0abdadcc6e1` by Patrick McHardy, which added the `nft_verdict` parsing path the UAF exploits, giving a roughly ten-year latent window under that heuristic. This is the form that post-pivot kernel CVEs take when they reach the top of the funnel. Most published CNA records since February 2024 do not reach it.

`d10119968d0e`: xfrm_user info leak (non-CVE). On 2026-04-06 Greg Kroah-Hartman signed off a one-paragraph commit on `net/xfrm/xfrm_user.c` titled "xfrm_user: fix info leak in build_report()". The body notes three bytes of padding in `struct xfrm_user_report` copied to userspace without being zeroed; the fix memsets the structure before populating fields. The commit carries a `Cc: stable` trailer and our `info_leak` signal. This commit fixes a memory disclosure in

the kernel networking stack. It ships in the stable tree. It has no CVE number and is not visible to any downstream CVE consumer. Across 22 repositories, 21,826 commits share this profile.

CVE-2023-29491: ncurses (distro-first). Debian shipped `DSA-5396` for ncurses on 2023-04-14; Red Hat issued RHSA-2023:3073 on 2023-05-16; NVD did not publish the CVE-2023-29491 record until 2023-04-13, but the CPE enrichment and CWE assignment arrived weeks later. The advisory stream carried the actionable fix before the NVD entry was consumable for automated inventories. Distro advisories are an earlier detection channel than NVD for a non-trivial share of the corpus, a resolution-side effect the timing analysis in §8 quantifies.

11 Discussion

CVE-based measurement undersells security work in the Linux base system by roughly 17×. In the kernel the ratio is 56:1. Any empirical study that treats CVE count as a proxy for security-work volume misses the majority of the stream.

A regime change for NVD-based corpora. Two events compound. The kernel CNA pivot in February 2024 re-routed kernel CVE issuance away from the NVD enrichment pipeline that most downstream consumers read. NIST's April 2026 shift to risk-prioritized enrichment formalizes the gap by ranking which records receive CPE, CVSS, and weakness data at all. Combined, they widen the distance between "listed in NVD" and "researchable from NVD." The kernel CNA has published 6,239 records since the pivot; 6,013 are still unclaimed by any downstream consumer in our corpus. Research corpora built against the NVD index alone will rot over the next several years, not because the defect stream dried up but because the label stream moved.

Distro streams are the earliest consistent detection surface. Red Hat ships fixes a median of 25 days after the upstream patch across 672 matched advisories, and Ubuntu ships at a median of 101 days across 534 matched advisories (`data/advisories/delay_summary.json`). NVD enrichment typically arrives weeks to months after either. For any downstream consumer who needs a consistent, machine-readable detection surface earlier than the NVD feed, distribution advisory streams are the right instrument. Treating NVD as the primary detector inverts the clock.

Author concentration is fragility. 3,413 distinct kernel security-signal authors sounds like breadth, but the power law in §9 is sharp: the top ten authors carry more security-signal commits than the bottom two thousand combined, and roughly half of all authors contribute one patch and do not return. A handful of individuals and a handful of automated reporters (syzbot on 3,310 kernel commits, OSS-Fuzz across most of the userspace repositories) do the load-bearing work. Burn-out, succession, and institutional documentation of what these people actually do day-to-day are implicit fragilities in the security posture of the base system.

The CVE number is a lagging indicator. 77.6% of the 1,051 dossiers with both timestamps land their fix before public disclosure. The bug is repaired in the source tree before the outside world hears about it; the CVE number, when it exists, arrives later, and distro advisories and stable backports arrive earlier still. A study that wants to measure "security activity" should mine commits first and CVEs second. Commit-keyed corpora preserve the ordering that the CVE record collapses.

Limitations. The 22 repositories in this paper are a curated sample of the Linux base system rather than a representative sample of all Linux software; extrapolation to language ecosystems or to vertical software outside the base-system perimeter is not warranted. Non-CVE security-signal classification is keyword-driven and will miss commits whose security relevance is implicit, so the 2.03% rate is best read as a lower bound on the true signal share. The classifier itself has a strict per-commit precision of 46.7% (Wilson 95% CI [30.2%, 63.9%], §4.1), so the 21,826 pool is correctly described as security-signal commits rather than security fixes. The headline rates are dominated by a small number of repositories (the kernel contributes most of the non-CVE volume; the aggregate CVE-assignment rate is driven by userspace runtimes with dense CVE practice); a leave-one-repo-out sensitivity analysis and an expanded repository sample are named future work rather than claims in this paper. Dossier construction is LLM-assisted, which means narrative accuracy rests on a pipeline whose error rate we have not yet measured against human adjudication on the *CVE* slice; a 200-sample audit with inter-rater κ on the scope flag and on the dossier's canonical fields is named future work. Container-inventory denominators are keyed to one Alpine-minimal image today, which bounds every package-prevalence claim to Alpine-minimal until the inventory is scaled across Debian and Ubuntu bases.

12 Conclusion

Of 23,122 security-relevant commits across 22 Linux base-system repositories, 5.6% acquire a CVE number. Of the kernel CNA's 6,239 records published since February 2024, 6,013 are still unclaimed by any downstream consumer in our corpus. Fix-first culture is real: 77.6% of 1,051 dossiers with both dates land their fix before public disclosure. Bug lifetimes run long at a median of 4.7 years across 888 reconstructable cases. Red Hat and Ubuntu ship distribution fixes 25 and 101 days after upstream, respectively, earlier than NVD enrichment arrives for most of the same records.

We release the commit-keyed corpus and the 1,418 full dossiers together under CC-BY-4.0 at `https://huggingface.co/datasets/mjbommar/linux-security-meanfield`, with the `2026-04-22 revision freeze` snapshot archived on Zenodo. CVE-based measurement captures roughly one in twenty security fixes. The rest are in the commits.

References

[1] Nikolaos Alexopoulos, Manuel Brack, Jan Peter Wagner, Tim Grube, and Max Mühlhäuser. How long do vulnerabilities live in the code? a large-scale empirical measurement study on FOSS vulnerability lifetimes. In *Proceedings of the 31st USENIX Security Symposium*, 2022.

[2] Afsah Anwar, Ahmed Abusnaina, Songqing Chen, Frank Li, and David Mohaisen. Cleaning the NVD: Comprehensive quality assessment, improvements, and analyses. *arXiv preprint arXiv:2006.15074*, 2020.

[3] Guru Prasad Bhandari, Amara Naseer, and Leon Moonen. CVEfixes: Automated collection of vulnerabilities and their fixes from open-source software. *arXiv preprint arXiv:2107.08760*, 2021.

[4] Yiran Cheng, Ting Zhang, Lwin Khin Shar, Shouguo Yang, Chaopeng Dong, David Lo, Shichao Lv, Zhiqiang Shi, and Limin Sun. VERCATION: Precise vulnerable open-source software version identification based on static analysis and LLM. *arXiv preprint arXiv:2408.07321*, 2025.

[5] Jiahao Fan, Yi Li, Shaohua Wang, and Tien N. Nguyen. A C/C++ code vulnerability dataset with code changes and CVE summaries. In *Proceedings of the 17th International Conference on Mining Software Repositories (MSR)*, pages 508–512. ACM, 2020.

[6] Zeyu Gao, Junlin Zhou, Bolun Zhang, Yi He, Chao Zhang, Yuxin Cui, and Hao Wang. Mono: Is your "clean" vulnerability dataset really solvable? exposing and trapping undecidable patches and beyond. *arXiv preprint arXiv:2506.03651*, 2025.

[7] Antonios Gkortzis, Dimitris Mitropoulos, and Diomidis Spinellis. VulinOSS: A dataset of security vulnerabilities in open-source systems. In *Proceedings of the 15th International Conference on Mining Software Repositories (MSR)*, pages 18–21. ACM, 2018.

[8] José Gonçalves, Tiago Dias, Eva Maia, and Isabel Praça. SCoPE: Evaluating LLMs for software vulnerability detection. *arXiv preprint arXiv:2407.14372*, 2024.

[9] Frank Li and Vern Paxson. A large-scale empirical study of security patches. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 2201–2215. ACM, 2017.

[10] Jiahuei Lin, Haoxiang Zhang, Bram Adams, and Ahmed E. Hassan. Vulnerability management in linux distributions: An empirical study on debian and fedora. *Empirical Software Engineering*, 28(47), 2023.

[11] Xiang Mei, Pulkit Singh Singaria, Jordi Del Castillo, Haoran Xi, Abdelouahab Benchikh, Tiffany Bao, Ruoyu Wang, Yan Shoshitaishvili, Adam Doupé, Hammond Pearce, and Brendan Dolan-Gavitt. ARVO: Atlas of reproducible vulnerabilities for open source software. *arXiv preprint arXiv:2408.02153*, 2024.

[12] Juefei Pu, Xingyu Li, Zhengchuan Liang, Jonathan Cox, Yifan Wu, Kareem Shehada, Arrdya Srivastav, and Zhiyun Qian. Patch-to-PoC: A systematic study of agentic LLM systems for linux kernel N-day reproduction. *arXiv preprint arXiv:2602.07287*, 2026.

[13] Bonan Ruan, Jiahao Liu, Weibo Zhao, and Zhenkai Liang. Vulzoo: A comprehensive vulnerability intelligence dataset. *arXiv preprint arXiv:2406.16347*, 2024.

[14] Various Authors. What do they fix? LLM-aided categorization of security patches for critical memory bugs. In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2026.

[15] Siqi Zhang, Minjie Cai, Mengyuan Zhang, Lianying Zhao, and Xavier de Carné de Carnavalet. The flaw within: Identifying CVSS score discrepancies in the NVD. In *International Conference on Cloud Computing Technology and Science (CloudCom'23)*, pages 185–192. IEEE, 2023.

A Release Manifest and Rebuild Path

The `2026-04-22 revision freeze` snapshot is a single tarball plus a live HuggingFace mirror. This appendix lists the on-disk layout, gives the ordered rebuild path, and pins the prompt, schema, and model under which the dossier agent ran.

A.1 Release layout

The release ships the following files under the snapshot root:

- `commits.parquet` — 23,863 commit-keyed rows covering the CVE-dossiered and non-CVE security-signal splits, keyed on `(repo, commit_sha)`.
- `hard_negatives.parquet` — 1,138 records that a naive Linux-kernel CPE query returned but that the scope audit ruled out of scope.
- `patches/<CVE>/*.patch` — 5,479 raw patch files preserved as artifacts for the CVE-dossiered split.
- `schema.json` — the JSON Schema the dossier agent emits against, pinned by SHA-256.
- `CITATION.cff` and `LICENSE` — CC-BY-4.0 licensing and a citation record keyed to the snapshot label.
- `README.md` — the dataset card, describing splits, schema, and known limitations.

A.2 Rebuild path

A clean rebuild runs seven steps in order from the repository root:

1. `./scripts/refresh_kernel_cna.sh`
2. `uv run python scripts/validate_dossier_invariants.py -strict`
3. `uv run python scripts/build_dossier_index.py`
4. `uv run python scripts/generate_meanfield_paper_tables.py`
5. `uv run python scripts/analyze_meanfield_story.py`
6. `uv run python scripts/export_meanfield_parquet.py`
7. `cd papers/meanfield && latexmk -pdf main.tex`

Step 2 is the release gate: a non-zero exit blocks every subsequent step. Steps 4 through 6 regenerate the tables, figures, and Parquet exports from frozen JSON artifacts; steps 1 and 3 refresh the upstream CNA view and the dossier index respectively.

A.3 Prompt and schema pinning

The dossier agent runs against two pinned files:

- `prompts/dossier_write_v2.md`
  SHA-256 `73153946eaabd2eb66218c8fd6436dfc830b85a840dd14c143a137150b0e5fc8`.
- `prompts/dossier_schema.json`
  SHA-256 `354b9cee1d6769ded6d21a5b81f93d18f04dc838b887fec8cd42303874d6d94a`.

Each per-CVE `run_manifest.json` records both SHA-256 values at issuance time, so a reader can detect prompt or schema drift without re-running the agent.

A.4 Pinned model

The agent runs the `codex` CLI against model `gpt-5.4`, configured via `~/.codex/config.toml`, with `model_reasoning_effort=medium`, `temperature=0.0`, and `seed=42`. A reader who re-runs the pipeline under different settings is expected to bump the snapshot label and record the delta, not overwrite the `2026-04-22 revision freeze` artifacts.

A.5 DOI and mirror

The Zenodo DOI is pending deposition against the `2026-04-22 revision freeze` snapshot and will be minted at submission time. The HuggingFace mirror is live at https://huggingface.co/datasets/mjbommar/linux-security-meanfield under the `2026-04-22 revision freeze` snapshot label; it carries the same Parquet files, patches, schema, and citation record as the tarball.