CIRCL
Computer Incident
Response Center
Luxembourg

The Missing Map: Crowdsourcing Better CPEs for Vulnerability Management with `cpe.gcve.eu`

GCVE-BCP-10, cpe-editor, public service and open-data workflow

Alexandre Dulaunoy, CIRCL

VulnOpticon 2026 – September 2026

CIRCL – Computer Incident Response Center Luxembourg
https://cpe.gcve.eu – https://gcve.eu/bcp/gcve-bcp-10/

TLP:CLEAR

Why CPE needs a better map

The story in 30 minutes

1. Problem

CPE connects vulnerabilities to products, but the naming layer is brittle.

2. Standard

GCVE-BCP-10 keeps CPE compatibility and adds stable registry records.

3. Tooling

`cpe-editor` and `cpe.gcve.eu` turn the model into reusable infrastructure.

4. Community

Crowdsourced proposals improve vendor, product, CPE and PURL mappings.

The goal is not to replace CPE. The goal is to make CPE data easier to curate, verify, reuse and synchronize.

Why this matters for vulnerability management

Every operational question needs product identity

- Is this vulnerability relevant to my inventory?
- Which vendor and product does the advisory really mean?
- Is this CPE an alias, a rename, a fork, or a deprecated entry?
- Can two tools agree on the same platform without manual glue?

Bad mappings become operational debt

- Missed vulnerabilities when product names do not match.
- False positives when broad CPEs are over-applied.
- Duplicate work across CSIRTs, vendors, scanners and SBOM tools.
- Unclear provenance when a mapping was imported, corrected or deprecated.

CPE is the quiet layer between vulnerability intelligence and asset reality. When it is wrong, everything downstream becomes less reliable.

The missing map

Vulnerability records and asset inventories often use different names for the same thing.

Vulnerability data	Missing map	Operational data
CVE/GCVE records, advisories, CPE ranges, affected products.	Vendor aliases, product renames, package names, PURL links, deprecated CPEs.	Asset inventories, SBOMs, exposure management, scanner findings, CTI platforms.

CPE.GCVE.EU is the public workspace to improve that map collectively.

Traditional CPE is "useful", but not enough

What already works

- Compact, widely understood platform naming.
- Existing CPE 2.3 strings remain central to matching.
- Many vulnerability tools already rely on CPE-compatible data.

What needs improvement

- The name often acts as the identifier.
- Aliases, synonyms and renames are difficult to track.
- Vendor mergers, splits and rebranding need lifecycle handling.
- Provenance, approval and metadata need a first-class place.

BCP-10 separates stable identity from display names, CPE-compatible strings and descriptive metadata.

GCVE-BCP-10

GCVE-BCP-10 in one sentence

A documented, reusable model for improving CPE data without breaking existing CPE-based tooling.

- Defines records for **vendors**, **products**, **CPE entries**, **metadata**, **relationships** and optional **proposals**.
- Keeps CPE-compatible strings for matching and interoperability.
- Adds deterministic UUIDv5 identifiers for durable vendor and product identity.
- Introduces structured relationships for synonyms, renames, equivalence and lifecycle changes.
- Makes registry synchronization and open-data reuse explicit.

Reference: GCVE-BCP-10, version 1.1, draft for public review, 2026-04-29.

Design principles

Compatibility first

Existing CPE 2.3 strings remain valid and continue to serve matching workflows.

Stable identity

Vendor and product records get UUIDs so identity survives naming changes.

Deterministic import

UUIDv5 makes repeated imports idempotent across instances.

Structured relationships

Synonyms, aliases, renames and equivalence become records, not tribal knowledge.

Namespaced metadata

Metadata such as `gcve:url` and `gcve:description` is attached cleanly.

Auditability

Submit, approve, create and update timestamps support review and synchronization.

The BCP-10 data model

Record	Purpose
Vendor	Stable identity for an organization, project or vendor name.
Product	Stable identity for a product, platform, application, hardware item or package.
CPE entry	Full CPE 2.3 URI plus parsed fields, linked to vendor and product UUIDs.
Metadata	Namespaced key/value assertions attached to a vendor or product.
Relationship	Typed links such as `synonym-of`, `renamed-to`, `equivalent-to`.
Proposal	Moderated change requests for missing or corrected information.

The model turns CPE curation into data that can be exchanged, reviewed and synchronized.

Deterministic UUIDs: stable enough to synchronize

Vendor identity

```
vendor_uuid = uuid5(
  VENDOR_UUID_NAMESPACE,
  normalize_token(vendor_name)
)
```

Product identity

```
product_uuid = uuid5(
  PRODUCT_UUID_NAMESPACE,
  f"{vendor}:{product}"
)
```

- Same normalized input produces the same UUID.
- Product identity is vendor-scoped to avoid product-name collisions.
- Imports can be repeated without generating random duplicate records.
- Names can evolve while identity remains stable.

This is essential for distributed registries, git datasets and local mirrors.

Relationships are first-class records

Real-world naming is messy

- A vendor changes name.
- A product is renamed or rebranded.
- A project has historical names, package names and marketing names.
- Different ecosystems describe the same component differently.

BCP-10 records the semantics

- `synonym-of`: alternative name for same identity.
- `renamed-to`: directional lifecycle change.
- `equivalent-to`: equivalent records in practice.
- `superseded-by`: deprecated record replaced by another.

Instead of losing context in comments or local scripts, curation decisions become reusable registry data.

Implementation and public service

From BCP to implementation

Standard

GCVE-BCP-10 defines the format and minimal compliance profile.

Open-source tooling

cpe-editor implements search, import, proposals, relationships and exports.

Online service

cpe.gcve.eu exposes the workflow publicly for users and contributors.

The important part: the standard, tooling and service exist together, so the model can be tested operationally rather than only discussed.

The source model

Upstream CPE data

- NVD CPE dictionary
- NVD CPE Match archive
- deprecated and replacement entries

Ecosystem mappings

- PURL → CPE mappings
- GCVE-enriched dumps (CSAF and others)
- CVE/GCVE/GHSA references

Human curation

- new vendor/product proposals
- edit and note proposals
- aliases, renames, relationships

The database is not a single import. It is a continuously curated merge of authoritative feeds, package-ecosystem mappings, vulnerability references and reviewed community input.

CPE.GCVE.EU: discover, moderate, distribute

Discover

Search canonical platform names, vendors, products, PURL and CPE URIs.

Moderate

Submit proposals, review improvements and preserve approval history.

Distribute

Reuse the data through API access, change history and dataset export.

- Designed for maintainers, reviewers and consumers of security metadata.
- Public interface for browsing vendors, products, CPE entries and approved changes.
- Proposal workflow for missing or incorrect data.

What can be improved by the crowd?

Low-friction contributions

- Propose a missing vendor or product.
- Add an official vendor URL or description.
- Link a product to a CPE URI or package URL.
- Report an incorrect or obsolete CPE entry.

Higher-value curation

- Merge aliases and spelling variants.
- Record renamed products and vendors.
- Add PURL/CPE mappings for open-source software.
- Validate mappings used by vulnerability records.

The project treats platform naming as community-maintained infrastructure, not as an opaque catalogue.

CPE Editor: reusable tooling

Implementation capabilities

- Import NVD CPE feeds and CPE match data.
- Import PURL-to-CPE mappings.
- Import CVE-to-CPE references from GCVE enriched dumps.
- Manage proposals, metadata and relationships.

Operational reuse

- Export portable `cpe-editor-dataset` bundles.
- Export deterministic git-friendly dataset trees.
- Run API tests and local instances.
- Integrate with local vulnerability and asset workflows.

Source code: `https://github.com/gcve-eu/cpe-editor`

The exchange shape is intentionally simple

```
{
  "format": "cpe-editor-dataset",
  "version": "1",
  "exported_at": "...",
  "counts": { ... },
  "vendors": [],
  "products": [],
  "cpes": [],
  "metadata": [],
  "relationships": [],
  "proposals": []
}
```

- Human-readable JSON structure.
- Directly aligned with BCP-10 records.
- Can be mirrored, reviewed and transformed.
- Suitable for local processing and public open-data publication.

The dataset[a] should be easy to inspect with standard tools.

[a] https://cpe.gcve.eu/dumps/

API and automation use cases

Query

Search vendors, products, CPE entries, proposals, relationships and metadata.

Enrich

Add stable vendor/product context to vulnerability records, asset data and SBOM items.

Synchronize

Pull updates into local mirrors, scanners, CTI platforms or exposure-management systems.

- Vulnerability enrichment: CVE/GCVE record → vendor/product/CPE context.
- Asset inventory: local software names → normalized CPE-compatible entries.
- SBOM bridge: PURL/package ecosystems → CPE-based vulnerability matching.

Operational workflows

Workflow: fixing a missing or wrong CPE mapping

1. Analyst searches `cpe.gcve.eu` for the vendor or product.
2. If missing or inaccurate, the analyst submits a proposal with context.
3. Reviewer validates the proposal, relationship or metadata assertion.
4. Approved change becomes visible in history, API and future exports.
5. Downstream tools synchronize the improved mapping.

The same improvement can benefit many organizations instead of remaining in one local spreadsheet or parser rule.

Workflow: from vulnerability record to affected asset

Without a maintained map

- Advisory naming does not match asset naming.
- Product rename creates duplicate identities.
- Open-source package has PURL but no CPE link.
- Analysts manually patch data per tool.

With BCP-10 + CPE.GCVE.EU

- Vendor/product have stable UUIDs.
- CPE entries remain compatible with existing matching.
- Relationships explain aliases and renames.
- PURL/CPE relationships connect supply-chain and vulnerability workflows.

How this fits the broader GCVE initiative

GCVE records need better product context

- Vulnerability records describe affected products.
- CPE/PURL mappings help records become actionable.
- Deterministic identifiers support federation and reproducible imports.

CPE data needs an open process

- Public registry for vendor and product curation.
- Open-source implementation for reuse and verification.
- Open-data exchange for synchronization across communities.

This is a practical example of GCVE's strategy: open standard, open tooling, open data, and decentralized contribution.

Contribution and adoption

Who benefits?

Producers and curators

- CSIRTs and vulnerability coordinators.
- Open-source maintainers and vendors.
- Researchers creating advisory data.
- Communities maintaining sector-specific platform catalogues.

Consumers and integrators

- Vulnerability management teams.
- Scanner and exposure-management vendors.
- SBOM and supply-chain tooling.
- CTI platforms and data pipelines.

Better CPE data is shared leverage: one correction can improve many downstream decisions.

How to contribute

1. Use `https://cpe.gcve.eu` to search existing vendors, products, CPEs and PURL relationships.
2. Submit proposals for missing or incorrect data with enough evidence to review.
3. Open issues or patches against `gcve-eu/cpe-editor` for tooling and API improvements.
4. Discuss BCP-10 improvements through the GCVE public review process.
5. Reuse the dataset and report where synchronization or matching breaks.

The most valuable contributions are often small: a vendor alias, a product rename, an official URL, or a correct PURL/CPE link.

What to improve next

Data quality

- More curated PURL/CPE relationships.
- More explicit deprecation and replacement links.
- More vendor and product metadata with provenance.

Operational adoption

- More local mirrors and integrations.
- Better feedback loops from scanners, SBOM tools and asset systems.
- Community-specific curation workflows for sectors and ecosystems.

BCP-10 is intentionally practical: the model should evolve with real curation and integration experience.

Takeaways

- CPE remains essential for vulnerability management, but the surrounding registry data needs to be more durable and auditable.
- GCVE-BCP-10 keeps CPE compatibility while adding stable IDs, metadata, relationships and synchronization semantics.
- `cpe-editor` is the open-source implementation; `cpe.gcve.eu` is the public collaborative service.
- Crowdsourcing CPE improvements turns local fixes into shared infrastructure.

The missing map can only be fixed collectively. GCVE provides the standard, tooling and open-data path to do it.

References

- GCVE-BCP-10: Improved Common Platform Enumeration for GCVE
  `https://gcve.eu/bcp/gcve-bcp-10/`
- CPE.GCVE.EU public service
  `https://cpe.gcve.eu/`
- cpe-editor open-source implementation
  `https://github.com/gcve-eu/cpe-editor`
- Introduction blog post: collaborative catalog for vendors, products, CPEs and PURLs
  `https://gcve.eu/2026/06/02/introducing-cpe.gcve.eu-a-collaborative-catalog-for-vendors-products-cpes-and-purls/`

Questions?

https://cpe.gcve.eu

https://gcve.eu