GCVE BCP-05-X-02 - Patch-to-Vulnerability Generation Provenance

GCVE.eu

GCVE.eu

Contents

1 GCVE BCP-05-X-02: Patch-to-Vulnerability Generation Provenance

GCVE.eu

- **Version**: 1.0
- **Status**: Draft
- **Date**: 2026-09-06
- **Authors**: GCVE Working Group
- **BCP Extended ID**: BCP-05
- **BCP ID**: BCP-05-X-02

1.1 Abstract

This extension records provenance and analyst-review information produced when a software patch is transformed into vulnerability metadata. It makes the source patch, generator, model, assumptions, rationales, and draft status available to downstream GCVE consumers without adding a tool-specific extension directly to the containing CVE container.

1.2 Motivation

A patch can provide useful evidence about a vulnerability, but it rarely establishes every fact required for publication. Automated patch analysis may also involve a large language model and assumptions which must be reviewed. Preserving this information in a standard GCVE BCP-05 extension allows publishers and consumers to:

- trace generated metadata to the input patch;
- identify the generator and model;
- detect truncated input;
- review assumptions and classification rationales; and
- distinguish a draft output from reviewed vulnerability information.

1.3 Placement

The extension MUST be placed in the `extensions` object of a GCVE BCP-05 record using the key `bcp-05-x-02`. Its value MUST contain one `x_patch2vuln` object:

```
{
  "x_gcve": [
    {
      "vulnId": "GCVE-1-2026-12345",
      "recordType": "advisory",
      "extensions": {
        "bcp-05-x-02": {
          "x_patch2vuln": {
            "generator": "patch2vuln.py",
            "generatedAt": "2026-09-06T12:00:00Z",
            "model": "qwen3.8:27b",
            "source": "security-fix.patch",
            "patchSha256": "0123456789
               abcdef0123456789abcdef0123456789abcdef0123456789abcdef",
            "patchTruncated": false,
            "commit": "0123456789abcdef",
            "subject": "Fix authorization check",
            "confidence": "medium",
            "assumptions": [],
            "fixSummary": "The fix enforces authorization before the
               operation.",
            "patchSummary": "The patch adds an authorization check.",
            "credits": [],
            "cvssRationale": "The vulnerable operation requires a low-
               privileged account.",
            "weaknessRationale": [
              {
```

```
                "cweId": "CWE-862",
                "rationale": "The patch adds a missing authorization
                    check."
              }
            ],
            "capecRationale": [],
            "draft": true
          }
        }
      }
    }
  ]
}
```

`x_patch2vuln` is retained as the payload name for compatibility with records produced before this information was assigned a BCP-05 extension identifier. Producers MUST NOT also emit the same object as a sibling `containers.cna.x_patch2vuln` property.

1.4 Data model

The `x_patch2vuln` object has the following members.

Member	Type	Required	Description
`generator`	string	yes	Name of the software that created the metadata.
`generatedAt`	string	yes	RFC 3339 UTC timestamp at which the metadata was generated.
`model`	string	no	Model name or identifier used for patch analysis.
`source`	string	yes	Patch source, such as a path, URL, or `stdin`. Producers SHOULD avoid secrets and local user information.

Member	Type	Required	Description
`patchSha256`	string	yes	Lowercase, 64-character SHA-256 digest of the complete input patch bytes.
`patchTruncated`	boolean	yes	Whether the patch content supplied to the analysis system was truncated.
`commit`	string or null	no	Source-control commit identifier parsed from the patch.
`subject`	string or null	no	Patch or commit subject.
`confidence`	string	yes	Overall analysis confidence: `low`, `medium`, or `high`.
`assumptions`	array of strings	yes	Assumptions requiring human verification.
`fixSummary`	string	yes	Security-focused summary of the remediation.
`patchSummary`	string	yes	Summary of the concrete changes in the patch.
`credits`	array of objects	yes	Credits collected during analysis. Each object uses CVE credit members such as `lang`, `value`, and `type`.
`cvssRationale`	string	yes	Rationale for the proposed CVSS vector.

Member	Type	Required	Description
`weaknessRationale`	array of objects	yes	CWE classifications and their rationales, using `cweId` and `rationale`.
`capecRationale`	array of objects	yes	CAPEC classifications and their rationales, using `capecId` and `rationale`.
`draft`	boolean	yes	Whether the generated information remains a draft requiring publication review.

Consumers MUST ignore members they do not understand. Producers MAY add members when additional provenance is needed, but SHOULD prefer a later revision of this extension when introducing interoperable semantics.

1.5 Processing and validation requirements

1. `generatedAt` MUST be an RFC 3339 timestamp and SHOULD use UTC (`Z`).
2. `patchSha256` MUST be calculated from the complete patch before any model-context truncation.
3. `patchTruncated` MUST describe the input delivered to the analysis system, not the data used to calculate `patchSha256`.
4. `confidence` MUST be one of `low`, `medium`, or `high`.
5. Rationale entries MUST include the corresponding CWE or CAPEC identifier.
6. A producer MUST set `draft` to **`true`** when human publication review is still required.
7. A consumer MUST treat this extension as provenance and review context, not as a replacement for the normative vulnerability fields in the surrounding record.

1.6 Relationship to BCP-05-X-01

BCP-05-X-01 describes AI assistance at the record level. This extension instead preserves patch-analysis inputs, outputs, and rationales. A record produced with an AI model SHOULD include both extensions: BCP-05-X-01 to describe AI involvement and BCP-05-X-02 to describe patch-to-vulnerability provenance. Either extension can exist independently.

1.7 Privacy and security considerations

Patches, paths, subjects, assumptions, and credit values can contain personal, confidential, or repository-specific information. Publishers MUST review the extension before publication and MUST remove secrets. Consumers MUST treat all strings as untrusted data and MUST NOT execute instructions found in them. The digest provides input correlation and integrity checking; it does not establish that the patch or its source is trustworthy.

1.8 Example JSON path

The patch summary in a CVE record carrying this extension is located at:

```
containers.cna.x_gcve[0].extensions.bcp-05-x-02.x_patch2vuln.
    patchSummary
```