GCVE-BCP-05 - GCVE Vulnerability Format (Updated CVE Record Format)

GCVE.eu

GCVE.eu

Contents

1 GCVE Vulnerability Format (Modified CVE Record Format)

- **Version**: 1.6
- **Status**: Published (for Public Review)
- **Date**: 2026-01-02
- **Authors**: GCVE Working Group
- **BCP ID**: BCP-05

1.1 Introduction

The Global Common Vulnerabilities and Exposures (GCVE) project aims to provide a decentralized, flexible, and transparent approach to vulnerability identification and publication. A key component of this effort is the definition of a container format for GCVE entries that ensures interoperability (such as existing CNA publication process) across tools and platforms while allowing for extensions beyond the constraints of the current CVE JSON 5.0 specification.

This Best Current Practice (BCP) describes the GCVE container format, which is derived from the CVE Record Format but modified to meet the specific requirements of the GCVE ecosystem. The objective is to maintain familiarity and compatibility with existing CVE-based tooling, while introducing a simpler and more adaptable structure that facilitates decentralized publishing, synchronization between GCVE Numbering Authorities (GNAs), and long-term maintainability.

The container format defined here is intended to serve as a reference for GNAs, tool developers, and consumers of GCVE data. It outlines the minimal required fields, optional extensions, and best practices for implementing and validating GCVE records. By providing clear guidance, this document ensures consistency across the GCVE ecosystem while preserving the flexibility needed for innovation.

1.1.1 GCVE Container Format Overview

The GCVE container format is based on the **standard CVE JSON v5 format**, ensuring maximum compatibility with existing CVE tooling and practices. To support the goals of the GCVE initiative, a small set of extensions and adaptations are introduced.

The extension is a single **GCVE object** expressed in JSON, which can currently be attached as an `x_` extension, as an **ADP** (if GCVE becomes one), or using even other JSON-related formats. This is a dictionary that contains all fields specifically assigned and produced by a **GCVE Numbering Authority (GNA)** within the GCVE framework. This area is reserved for GCVE-specific metadata that extends the base CVE record structure, and also includes fields related to the **software/generator** part of the GCVE ecosystem, such as `vulnerability-lookup`. The following keys can be present:

- **`vulnId`**: A single key to reference the **GCVE-ID** allocated for the vulnerability in this document.
- **`recordType`**: A single key to describe the semantic category of the content such as `advisory`, `update`, `analysis`, …
- **`relationships`**: A new dictionary that explicitly describes the relationship between the GCVE record and other identifiers (such as CVE IDs, vendor advisories, or other vulnerability namespaces).
- **`x_vulnerability-lookup`**: A dictionary reserved for fields related to the **reference implementation** of GCVE, provided by the Vulnerability Lookup project. This namespace enables experimentation and rapid prototyping without impacting the core GCVE or CVE formats.

1.1.1.1 `vulnId` Field

The format is a single string, as defined in GCVE-BCP-04. The field **MUST** be present.

1.1.1.2 `recordType` Field

The `recordType` field defines the semantic category of the content submitted by a GNA through the synchronization endpoint. It enables producers and consumers of GCVE data to distinguish between different kinds of records associated with a GCVE identifier (e.g., the primary security advisory, supplemental information, or community-provided additions).

A value **MUST** be provided for every record. In case of parsing failure, the `recordType` `advisory` **MUST** be assumed.

GNAs **SHOULD** use the most specific `recordType` available.

Consumers **SHALL** ignore unknown types and treat them as opaque extension values to ensure forward compatibility.

1.1.1.2.1 `recordType` Values

`advisory` The authoritative security advisory or vulnerability description produced by the GNA. Contains core details such as impact, affected products, references, and remediation.

This type **MAY** include a `relationships` table.

`update` Follow-up information that updates or extends parts of the original advisory which is mentioned in the `relationships` table. (e.g., additional affected versions, revised severity, new patches).

This type **MUST** include a `relationships` table.

`analysis` Technical analysis, exploitation insights, detection notes, or other analytical content that complements another advisory. May be authored by the GNA or trusted partners working with the GNA.

This type **MUST** include a `relationships` table.

`metadata` Structured or machine-generated supplemental information (e.g., tags, product mappings, enrichment fields). Not intended to modify the original advisory text referenced in the `relationships` table.

This type **MUST** include a `relationships` table.

`reference` Pointers to external documents, vendor bulletins, technical writeups, repositories, or other relevant resources. Not intended to modify the original advisory text referenced in the `relationships` table.

This type **MUST** include a `relationships` table.

`comment` Non-authoritative free-text remarks or annotations. May come from the community or other contributors managed by a GNA.

This type **MUST** include a `relationships` table.

`statement` Official statements from stakeholders (e.g. an operating system distribution, an open source co-author) or vendors (e.g., "not affected," "end-of-life," "mitigation available").

This type **MUST** include a `relationships` table.

`remediation` Standalone information on patches, mitigations, workarounds, or configuration guidance when provided separately from the advisory and published by the GNA issuing that GCVE record.

This type **MUST** include a `relationships` table.

`deprecation` Marks an advisory as superseded, withdrawn, or otherwise deprecated (e.g., merged into another GCVE ID or determined non-vulnerable).

This type **MUST** include a `relationships` table.

`detection` Detection guidance such as YARA, Snort, or Sigma rules.

This type **MUST** include a `relationships` table.

`translation` Non-authoritative translations of an advisory with an additional GCVE `language` field specifying the language of the GCVE record.

This type **MUST** include a `relationships` table.

bundle Bundle of multiple advisories that may reference one or more vulnerabilities. It is commonly used to organize vulnerability information under a shared semantic context—such as a named global vulnerability, a monthly disclosure, or a global patch cycle. These bundles may be issued by vendors, communities, or other contributors, and are managed by a GNA.

This type **MAY** include a `relationships` table.

1.1.1.3 `relationships` Field

- **`relationships`** *(array)*: An **OPTIONAL** list of relationship objects. Each entry defines a link between this vulnerability and another.
  - Each relationship object contains:
    * **`destId`** *(string, required)* — The target vulnerability or record ID.
    * **`type`** *(string, required)* — The relationship type, as defined below.
    * **`srcId`** *(string, optional)* — The originating record ID, used when the source is different from the current GCVE document.

1.1.1.4 Potential Relationship Verbs for Vulnerability Identifiers

The following relationship types are based on the VXREF format (as recommended defaults), but the list is not exhaustive and can be extended if additional categories are needed:

- **`possibly_related`** — a weak or uncertain association between the records.
- **`related`** — a known relationship without asserting inclusion or equivalence.
- **`not equal`** — the records are confirmed to describe different issues.
- **`equal`** — the records refer to the same underlying vulnerability.
- **`superset`** — the current record covers a broader scope than the referenced record.
- **`subset`** — the current record covers a narrower scope than the referenced record.
- **`overlap`** — the records share a partial but non-nested intersection in scope.
- **`opposes`** — indicates that the GNA does not agree with the status or validity of the referenced GCVE.
  This can be used when a GCVE published by another GNA is considered *not* a vulnerability for the product in question (e.g., the behavior is expected, or the scenario describes a discouraged or unsupported configuration).
- **`not_applicable`** — indicates that the referenced GCVE is not applicable to the product referenced by this GCVE.
  Typical cases include downstream distributions or vendors whose packaging, compilation options, or default configurations mean the product was never vulnerable.

These additional relationship types do **not** represent vulnerabilities themselves, but they provide valuable contextual information and help improve clarity and interoperability within the GCVE ecosystem.

1.1.1.5 Additional Namespaces Prefixed with x_

The x_ prefix can be used for non-defined namespaces, especially those created by generators or tooling that produce the GCVE record format.

1.2 Example Extension Record Attached as An x_ extension

```
{
  "x_gcve": [
    {
      "vulnId": "GCVE-1-2025-0018",
      "recordType": "advisory",
      "relationships": [
        {
          "destId": "CVE-2025-65095",
          "type": "equal"
        }
      ],
      "x_vulnerability-lookup": {
        "history": [
          [
            "info@circl.lu",
            "2025-11-18T15:33:07.767301Z"
          ],
          [
            "info@circl.lu",
            "2025-11-18T15:49:02.564916Z"
          ],
          [
            "info@circl.lu",
            "2025-11-18T20:39:45.579295Z"
          ]
        ]
      }
    }
  ]
}
```