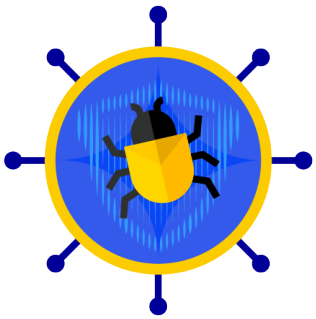
GCVE BCP-05-X-01 - AI-Assisted Vulnerability Information Annotation

GCVE.eu

GCVE.eu

Contents

1 GCVE BCP-05-X-01: AI-Assisted Vulnerability Information Annotation



GCVE.eu

- **Version**: 1.1
- **Status**: Published
- **Date**: 2026-05-18
- **Authors**: GCVE Working Group
- **BCP Extended ID**: BCP-05
- **BCP ID**: BCP-05-X-01

1.1 Abstract

This document defines an extension to GCVE BCP-05 to support the annotation of vulnerability records where Artificial Intelligence (AI) or automated processing has been used during their creation, enrichment, or analysis.

The objective is to provide transparency, traceability, and classification of AI-assisted contributions within vulnerability information, enabling consumers to assess trust, provenance, and review levels.

1.2 Scope

This extension applies to any GCVE record conforming to BCP-05 where:

- AI/ML models contributed to content generation, transformation, or classification
- Automated systems assisted human analysts
- Content was partially or fully generated by machine learning systems

This extension is optional but RECOMMENDED when such processing occurs.

If no AI or automated processing was involved, producers MAY explicitly state this by setting `ai_level` to `none`. In such cases, no model or review metadata is expected.

1.3 Extension Identifier

The extension identifier SHALL follow the GCVE BCP extension naming convention:

```
GCVE BCP-05-X-01
```

1.4 Data Model

1.4.1 Field Location

The AI annotation MUST be attached at one of the following levels:

- record-level: applies to the entire GCVE entry
- field-level: applies to specific fields within the record

The extension SHALL be embedded under:

```
{
  "x_gcve": [
    {
      "extensions": {
        "bcp-05-x-01": {
          "...": "..."
        }
      }
    }
  ]
}
```

1.4.2 Structure

```
{
  "bcp-05-x-01": {
    "ai_annotations": [
      {
        "scope": "record | field",
        "gna_source":  "integer",
        "field_name": "string (optional if scope=record)",
        "tags": ["string"],
        "description": "string",
        "ai_level": "none | assisted | augmented | generated",
        "review_status": "none | partial | full",
        "models": [
          {
            "name": "string",
            "version": "string (optional)",
            "provider": "string (optional)",
            "source": "ollama | huggingface | local | other",
            "identifier": "string (optional)",
            "url": "string (optional)"
          }
        ]
      }
    ]
  }
}
```

1.4.3 Conditional Field Requirements

The `review_status` and `models` fields are conditional and depend on the value of `ai_level`.

If `ai_level` is `none`:

- `review_status` MUST be omitted
- `models` MUST be omitted
- `tags` MAY be omitted or MAY contain taxonomy-aligned labels indicating that no AI assistance was used
- `description` MAY be used to explain the absence of AI or automated processing

If `ai_level` is one of `assisted`, `augmented`, or `generated`:

- `review_status` SHOULD be present
- `models` SHOULD be present when the model information is known
- `models` MAY be omitted when the model information is unavailable, unknown, or not applicable

- `tags` SHOULD be present to describe the type of AI-assisted processing

This distinction avoids implying that model or review metadata exists when no AI-assisted processing occurred.

1.5 Field Definitions

1.5.1 scope

Defines the applicability of the AI annotation.

Allowed values:

- `record`: applies to the entire vulnerability record
- `field`: applies to a specific field, such as `description`, `references`, or `analysis`

1.5.2 field_name

Specifies the affected field when `scope` is `field`.

Examples:

- `description`
- `title`
- `references`
- `analysis`

The `field_name` field MUST be present when `scope` is `field`.

The `field_name` field MUST be omitted when `scope` is `record`.

1.5.3 tags

The `tags` field is an array of classification labels describing the type and nature of AI-assisted processing applied to the vulnerability information.

Implementations are STRONGLY RECOMMENDED to reuse existing, well-defined taxonomies instead of defining ad-hoc or free-form tags. This improves interoperability, consistency, and machine-readability across GCVE producers and consumers.

In particular, the following MISP taxonomies SHOULD be preferred when applicable:

- AI Bias Terminology
- AI Computer Assisted
- AI Safety Benchmark

These taxonomies provide structured vocabularies to describe:

- The type of AI assistance, such as generation, classification, or summarization
- The level and nature of automation or augmentation
- Potential biases, risks, or safety considerations in AI-generated outputs

Tags derived from these taxonomies SHOULD follow their canonical naming and namespace conventions.

Example:

```
{
  "tags": [
    "ai-computer-assisted:llm-generated",
    "ai-computer-assisted:classification",
    "ai-bias:potential-hallucination"
  ]
}
```

Free-form tags MAY still be used when:

- No suitable taxonomy entry exists
- Experimental or domain-specific annotations are required

However, such tags SHOULD:

- Be clearly namespaced, for example `ai:custom-*`
- Avoid conflicting with existing taxonomy vocabularies
- Be documented for downstream consumers

Producers SHOULD prioritize taxonomy-aligned tagging whenever possible to ensure consistency across GCVE records.

1.5.4 description

Free-text description of the AI-assisted operation.

The `description` field SHOULD explain what role AI or automated processing played in the creation, enrichment, transformation, or analysis of the vulnerability information.

Examples:

- `The vulnerability description was summarized from vendor-provided text using an LLM and then reviewed by a human analyst.`
- `The affected product list was normalized using an automated classification system.`
- `No AI or automated processing was used for this record.`

1.5.5 gna_source

Identifies the GCVE Numbering Authority (GNA) responsible for producing, publishing, or asserting the AI annotation.

The `gna_source` value MUST be the numeric identifier of the GNA that performed or contributed the AI annotation.

The `gna_source` field MUST be represented as an integer value.

The `gna_source` field SHOULD be present for every AI annotation to support provenance, traceability, and accountability across decentralized GCVE producers.

Consumers SHOULD treat `gna_source` as a provenance signal indicating the source of the AI annotation, not as a guarantee of correctness.

1.5.6 ai_level

Defines the level of AI involvement.

Allowed values:

- `none`: no AI or automated processing was used
- `assisted`: AI or automation assisted a human analyst, but the human analyst remained the primary author or decision-maker
- `augmented`: AI or automation materially enriched, transformed, classified, or summarized the vulnerability information
- `generated`: AI generated the relevant content with limited or no human-authored input

When `ai_level` is `none`, the `review_status` and `models` fields MUST be omitted.

1.5.7 review_status

Indicates the level of human validation applied to AI-assisted content.

Allowed values:

- `none`: no human review was performed
- `partial`: some human review was performed
- `full`: the AI-assisted output was fully reviewed by a human analyst

The `review_status` field MUST be omitted when `ai_level` is `none`.

The `review_status` field SHOULD be present when `ai_level` is `assisted`, `augmented`, or `generated`.

1.5.8 models

List of AI models involved in the process.

The `models` field MUST be omitted when `ai_level` is `none`.

The `models` field SHOULD be present when `ai_level` is `assisted`, `augmented`, or `generated` and the model information is known.

Each model object MAY include the following fields:

- `name`: name of the model
- `version`: version of the model, if known
- `provider`: organization or project providing the model, if applicable
- `source`: source or execution environment of the model
- `identifier`: model identifier, registry name, digest, local identifier, or other stable reference
- `url`: URL identifying the model, model card, registry entry, project page, or documentation

Allowed values for `source`:

- `ollama`
- `huggingface`
- `local`
- `other`

The `url` field is OPTIONAL but RECOMMENDED when `source` is `other`, as it can help consumers identify the model or service used.

The `url` field MAY also be used with other `source` values when it provides a stable reference to a model card, registry page, documentation page, or source repository.

1.6 Examples

1.6.1 Record-Level Annotation with AI Assistance

```
{
  "x_gcve": [
    {
      "extensions": {
        "bcp-05-x-01": {
          "ai_annotations": [
            {
              "scope": "record",
              "gna_source": 1,
              "tags": [
                "ai-computer-assisted:llm-generated",
                "ai-computer-assisted:summarization"
              ],
              "description": "The vulnerability description was
                  summarized from vendor-provided advisory text using an
                   LLM and then reviewed by a human analyst.",
              "ai_level": "assisted",
              "review_status": "full",
              "models": [
                {
                  "name": "example-llm",
                  "version": "1.0",
                  "provider": "Example Provider",
                  "source": "other",
                  "identifier": "example-llm-1.0",
                  "url": "https://example.org/models/example-llm-1.0"
                }
              ]
            }
          ]
        }
      }
    }
  ]
}
```

1.6.2 Field-Level Annotation

```
{
  "x_gcve": [
    {
      "extensions": {
        "bcp-05-x-01": {
          "ai_annotations": [
            {
              "scope": "field",
              "gna_source": 1,
              "field_name": "description",
```

```
                "tags": [
                  "ai-computer-assisted:summarization"
                ],
                "description": "The description field was summarized from
                    a longer vendor advisory using an automated system
                   and partially reviewed by a human analyst.",
                "ai_level": "augmented",
                "review_status": "partial",
                "models": [
                  {
                    "name": "local-summary-model",
                    "version": "2026-01",
                    "source": "local",
                    "identifier": "sha256:exampledigest"
                  }
                ]
              }
            ]
          }
        }
      }
    ]
}
```

1.6.3 Explicit No-AI Annotation

```
{
  "x_gcve": [
    {
      "extensions": {
        "bcp-05-x-01": {
          "ai_annotations": [
            {
              "scope": "record",
              "gna_source": 1,
              "tags": [
                "ai-computer-assisted:none"
              ],
              "description": "No AI or automated processing was used
                  for this record.",
              "ai_level": "none"
            }
          ]
        }
      }
    }
  ]
}
```

In this example, `review_status` and `models` are intentionally omitted because `ai_level` is set to `none`.

1.7 Security and Trust Considerations

Consumers SHOULD evaluate AI-generated or AI-assisted content carefully.

AI-assisted vulnerability information may contain errors, hallucinations, incomplete analysis, misleading classifications, or incorrect references. Producers SHOULD disclose the level of AI involvement and the level of human review whenever AI-assisted processing materially contributed to the record.

Consumers SHOULD treat `ai_level`, `review_status`, `tags`, and `models` as provenance and trust signals, not as guarantees of correctness.

When the `url` field is used to identify a model or service, consumers SHOULD treat the URL as untrusted input. Implementations SHOULD avoid automatically fetching URLs without appropriate validation, filtering, and security controls.

1.8 Interoperability Considerations

This extension is backward-compatible with BCP-05. Consumers that do not support this extension can safely discard it.

Producers SHOULD use taxonomy-aligned tags whenever possible to improve interoperability across GCVE records.

Consumers SHOULD tolerate unknown fields inside the extension to allow future evolution of the data model.

Consumers SHOULD use `gna_source` to distinguish the GNA responsible for the AI annotation from other producers, enrichers, or downstream redistributors of the GCVE record.

Consumers SHOULD tolerate missing `gna_source`.

Consumers SHOULD also tolerate missing `models` fields when `ai_level` is `assisted`, `augmented`, or `generated`, as some producers may not have access to complete model metadata.

When `ai_level` is `none`, consumers SHOULD NOT expect `review_status` or `models` to be present.