Beyond CVEs: Mastering the Landscape with Vulnerability-Lookup

VSS 2025

https://www.vulnerability-lookup.org

Alexandre Dulaunoy - alexandre.dulaunoy@circl.lu Cedric Bonhomme - cedric.bonhomme@circl.lu

July 14, 2025

CIRCL https://www.circl.lu

TLP:CLEAR

Origin of the project

Who is behind Vulnerability-Lookup?

Vulnerability-Lookup[1] is an Open Source project led by **CIRCL**.
It is co-funded by **CIRCL** and the **European Union**[2].
Used by many organisations including CSIRTs and ENISA (EUVD).
A reference implementation to **GCVE** standards.

vulnerability
-lookup

[1] https://www.vulnerability-lookup.org
[2] https://github.com/ngsoti

Origin

- `cve-search`[3] is an open-source tool initially developed in late 2012, focusing on maintaining a **local** CVE database.
- `cve-search` is widely used as an **internal** tool.
- The design and scalability of cve-search are limited. Our operational public instance at https://cve.circl.lu has reached a hard limit of 20,000 queries per second.
- Vulnerability sources have **diversified**, and the **NVD CVE is no longer the sole source** of vulnerability information.

[3] https://github.com/cve-search/cve-search

Initial Challenges

- **Volume of data:** Handling a substantial dataset and heavy network traffic, currently over 1,360,500 security advisories and more than 90,000 sightings[4].
- **Flexibility:** Balancing ongoing development with legacy issues while designing a future-proof architecture. It's complex and yes, sometimes chaotic[5].
- **Robustness:** Validating data even when external entities don't comply with their own JSON schemas. It's not always pretty.
- **Fast lookup:** Rapidly correlating identifiers across **diverse sources**, including unpublished advisories.

[4] The first sighting on Exploit-DB dates back 26 years.
[5] We enjoy challenges, especially when they lead to practical solutions.

Ongoing Challenges and Development

- **CPE fragmentation:**[6] Tackling the fragmentation of CPEs (e.g., `cpe:/a:oracle:java` vs. `cpe:/a:sun:java`) by introducing *Organizations* as unified containers.
- **CVD process:** Building an open-source tool that fully supports the Coordinated Vulnerability Disclosure (CVD) process.[7]
- **Vulnerability numbering:** Enabling a new distributed approach through the Global CVE Allocation System.[8]
- **Scoring vulnerabilities:** Aggregating a large volume of observations from diverse advisory types to improve vulnerability scoring.

[6] Well, another mess to clean up!
[7] Aligned with NIS 2 and the Cyber Resilience Act.
[8] `https://gcve.eu`

Current Sources in Vulnerability-Lookup

- **CISA Known Exploited Vulnerability** (HTTP)
- **NIST NVD CVE** (API 2.0)
- **CVEProject - cvelist** (Git submodule)
- **Fraunhofer FKIE** (Git submodule)
- **Cloud Security Alliance - GSD** (Git submodule)
- **GitHub Advisory DB** (Git submodule)
- **PySec Advisory DB** (Git submodule)
- **CSAF 2.0** (HTTP CSAF)

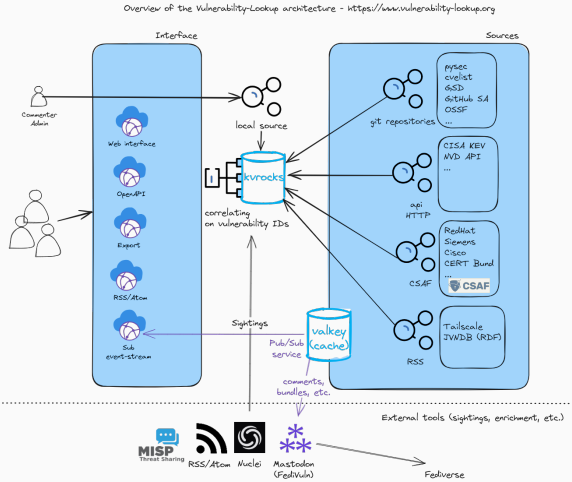
  CERT-Bund, Cisco, Siemens, Red Hat, Microsoft, NCSC-NL, CISA, etc.
- **VARIoT** (API)
- **Japan - JVN DB** (HTTP)
- **Tailscale** (RSS)
- **GCVE.eu all GNA sources**
- **CWE, CAPEC, MITRE EMB3D or KEV**
- **CNVD and growing...**

Open Data Initiative: Regular JSON dumps published[9].

[9] https://vulnerability.circl.lu/dumps/

Design and Implementation

Vulnerability-Lookup High-Level Architecture



TLP:CLEAR

Extended API

```
$ curl -s https://vulnerability.circl.lu/api/vulnerability/last/csaf_redhat/10 | jq .[2].document.title
"Red Hat Security Advisory: Red Hat Ceph Storage 6.1 security and bug fix update"

$ curl -s https://vulnerability.circl.lu/api/vulnerability/last/csaf_redhat/10 | jq .[2].vulnerabilities[0].cve
"CVE-2021-4231"
```

- **Documented API** (OpenAPI): https://vulnerability.circl.lu/api
- Pagination and filtering by source
- CPE search by vendor and product name
- **Many endpoints available via RSS and Atom**[10]

[10] https://www.vulnerability-lookup.org/documentation/feeds.html

Empowering the Community

Crowd-Sourced Threat Intelligence

- **Bundles:** Group similar vulnerabilities and aggregate sightings for easier tracking.
- **Comments:** Additional context such as PoCs, remediations, related insights.
- **Tags:** Use the MISP Vulnerability Taxonomy to annotate comments[11]. Example:

```
vulnerability:information=remediation
```

- **Sightings:** Report real-world observations of vulnerabilities, including metadata like timestamps and sources.


```
{
  "uuid": "f9ec8b2c-2ceb-4c05-b052-264b51c6a3ee", "vulnerability_lookup_origin": "1a89b78e-f703-45f3-bb86-59eb712668bd",
  "author": "9f56dd64-161d-43a6-b9c3-555944290a09", "creation_timestamp": "2025-04-17T19:14:32.000000Z",
  "vulnerability": "CVE-2025-32433",
  "type": "exploited",
  "source": "https://gist.github.com/numanturle/b7333fb02a4ee3618995babc9b62c507"
}
```


[11] https://www.misp-project.org/taxonomies.html#_vulnerability_3

Types of Sightings

Type	Description	Negative/Opposite
`seen`	The vulnerability was mentioned, discussed, or observed by the user.	-
`confirmed`	The vulnerability has been verified by an analyst.	X
`exploited`	The vulnerability was actively exploited and observed by the user reporting the sighting.	X
`patched`	The vulnerability was successfully mitigated or patched by the user reporting the sighting.	X

Table 1: Types of vulnerability sightings

Automated Sightings: Tools and Sources

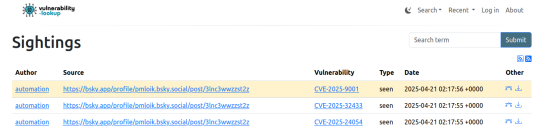
Automatically gathering crowd-sourced intelligence without requiring direct user contributions to our platform.

- **Social Platforms:** Fediverse, Bluesky
- **Threat Intelligence Tools:** MISP, Nuclei
- **Content Feeds:** RSS/Atom, curated web pages, GitHub Gist
- **Specialized Projects:** ShadowSight, ExploitDBSighting
- **Community Contributions:** Passive signals and indirect data enrichment

Scoring Vulnerabilities

Sightings Detection Rate and Types of Sightings

- A high rate of sightings (type *seen*) often correlates with high or critical severity vulnerabilities[12].
- Early sightings of type *exploited* (e.g., proof-of-concept code) or *confirmed* (e.g., detection templates for tools like Nuclei) can signal emerging threats.
- Sightings can sometimes be detected **before any official advisory is published**.

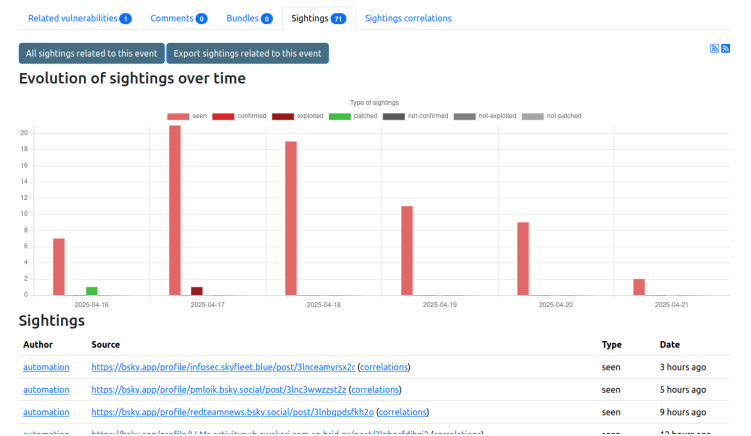


Author	Source	Vulnerability	Type	Date	Other
automation	https://bsky.app/profile/pmloik.bsky.social/post/3linc3wwzzst2z	CVE-2025-9001	seen	2025-04-21 02:17:56 +0000	
automation	https://bsky.app/profile/pmloik.bsky.social/post/3linc3wwzzst2z	CVE-2025-32433	seen	2025-04-21 02:17:55 +0000	
automation	https://bsky.app/profile/pmloik.bsky.social/post/3linc3wwzzst2z	CVE-2025-24054	seen	2025-04-21 02:17:55 +0000	

- Continuous exploitation patterns are frequently observed through sources like The Shadowserver Foundation or MISP.

[12] Don't underestimate the hype surrounding some vulnerabilities.

Early PoC (erlang / otp)

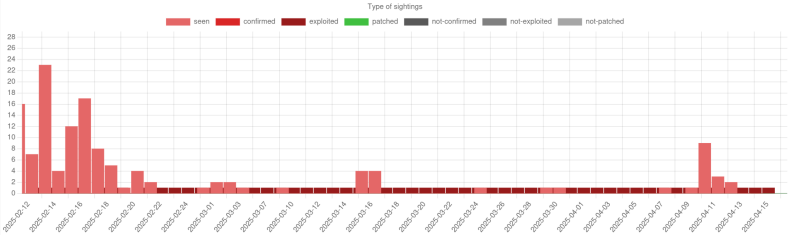


https://vulnerability.circl.lu/vuln/CVE-2025-32433#sightings

TLP:CLEAR

Continuous Exploitations (Palo Alto Networks / Cloud NGFW)

Evolution of sightings over time



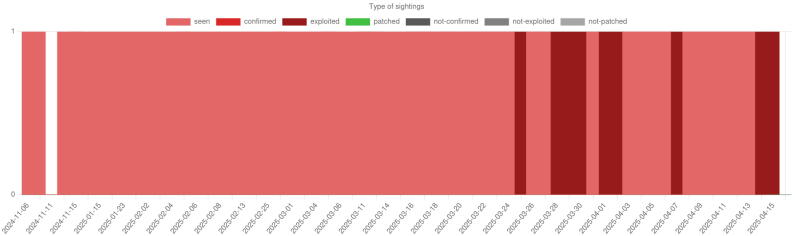
Sightings

Author	Source	Type	Date
automation	The Shadowserver (honeypot/exploited-vulnerabilities) - (2025-04-16) (correlations)	exploited	1 day ago
automation	https://bsky.app/profile/christopherkunz.bsky.social/post/3lmu2zatyx22z (correlations)	seen	2 days ago
automation	https://chaos.social/users/christopherkunz/statuses/114340622271163262 (correlations)	seen	2 days ago
automation	The Shadowserver (honeypot/exploited-vulnerabilities) - (2025-04-15) (correlations)	exploited	2 days ago

https://vulnerability.circl.lu/vuln/CVE-2025-0108#sightings

Continuous Exploitations (D-Link / DNS-320)

Evolution of sightings over time



Sightings

Author	Source	Type	Date
automation	The Shadowserver (honeypot/exploited-vulnerabilities) - (2025-04-16) (correlations)	exploited	1 day ago
automation	The Shadowserver (honeypot/common-vulnerabilities) - (2025-04-16) (correlations)	seen	1 day ago
automation	The Shadowserver (honeypot/common-vulnerabilities) - (2025-04-15) (correlations)	seen	2 days ago
automation	The Shadowserver (honeypot/exploited-vulnerabilities) - (2025-04-15) (correlations)	exploited	2 days ago

https://vulnerability.circl.lu/vuln/CVE-2024-10914#sightings

Last Month's Most Sighted Vulnerabilities

	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
CVE-2025-29927					3	11	54	42	20	15	7	10	1	3	1	1	4	4	1	2	1		2	1					1	1		
CVE-2025-22457																		39	38	11	12	16	8	6	5	13	3	4	3	4		14
CVE-2025-24813	13	15	12	13	8	3	2	11	2	1		1	1	3	5	7	7	4		2	1		1	2			1					
CVE-2025-1974								5	24	11	25	7	8	1	5	6	2	7							1							
CVE-2025-2825										2	10	7	2	2	11	9	12	7	2	2	2	3	6		5	3	1		1	3		
CVE-2025-29824																							12	29	11	4	2	1	4	2	3	14
CVE-2025-2783									1	27	15	12	8	7	2		1	1	1													
CVE-2025-30066	12	15	14	3	4	2	1	6	2	1	1				2								1		1							
CVE-2025-24200															3	3	4	3	1	1		3	1		12	30						
CVE-2017-18368	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2		
CVE-2015-2051	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2		
CVE-2025-30406																		1	2				2	3	6	2	2		8	14	3	14
CVE-2025-0108	1	5	5	1	1	1	1	1	1	1	2	1	1	1	1	1	3	1	1	1	1	1	1	1	2	1	2	3	11	3		

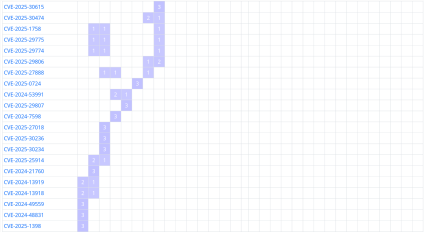
- **CVE-2025-22457:** Ivanti / Connect Secure — Severity: 10.0 (Critical)
- **CVE-2025-29927:** Vercel / Next.js — Severity: 9.1 (Critical)

Other Examples

Vulnerability	Product	Sighting count	EPSS	Severity
CVE-2025-29927	next.js	167	89.24% (0.99521)	9.1
CVE-2025-24813	Apache Tomcat	128	93.55% (0.99827)	9.2
CVE-2024-4577	PHP	190	94.38% (0.99961)	9.8
CVE-2025-0282	Connect Secure	243	90.87% (0.99618)	9.0
CVE-2024-55591	FortiOS	126	92.79% (0.99756)	9.8
CVE-2024-10914	D-Link DNS-320	81	93.73% (0.9985)	9.2
CVE-2020-21650	Myucms	57	2.48% (0.83998)	9.1

Table 2: Top vulnerabilities from our April 2025 report, based on sightings and scoring data.

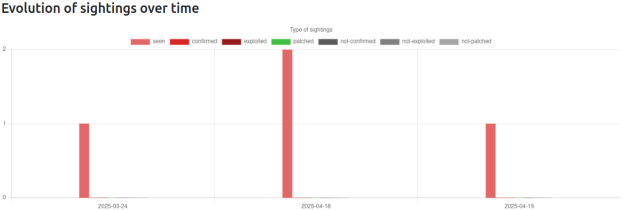
Least Sighted Vulnerabilities in the Last Month



- **Low-sighting outliers offer valuable intel**, even if absent from EPSS or predictive models.
- Particularly relevant in low-noise sources (e.g., MISP, private Telegram channels).
- Often rated low/medium by CVSS and have low EPSS scores.
- Trend highlights EPSS's dependence on public threat intel feeds.

Tracking the Exploitability of Vulnerabilities Prior to Public Disclosure

- **Google / Android:** https://vulnerability.circl.lu/vuln/CVE-2024-43093#sightings
- **Speedify VPN (macOS):** https://vulnerability.circl.lu/vuln/CVE-2025-25364#sightings
- **SourceCodester:** https://vulnerability.circl.lu/vuln/CVE-2025-3821#sightings
  - Low visibility, no EPSS score, few sightings



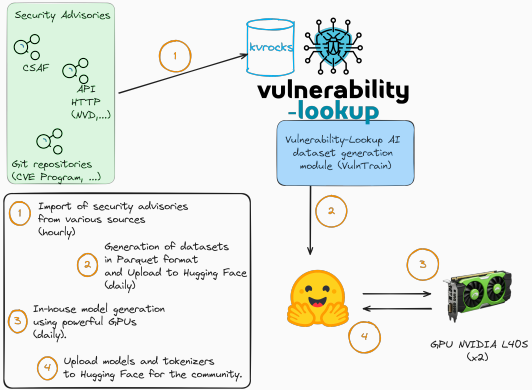
Sightings

Author	Source	Type	Date
automation	https://infosec.exchange/users/dragonjar/statuses/114364291565132421 (correlations)	seen	1 day ago
automation	https://bsky.app/profile/r-netsec.bsky.social/post/3ln4hb7anxx2q (correlations)	seen	2 days ago
automation	https://bsky.app/profile/r-netsec-bot.bsky.social/post/3ln4arcbktd2z (correlations)	seen	2 days ago
automation	https://infosec.exchange/users/threatcodex/statuses/114217935883108579 (correlations)	seen	27 days ago

TLP:CLEAR

Toward Practical AI Applications

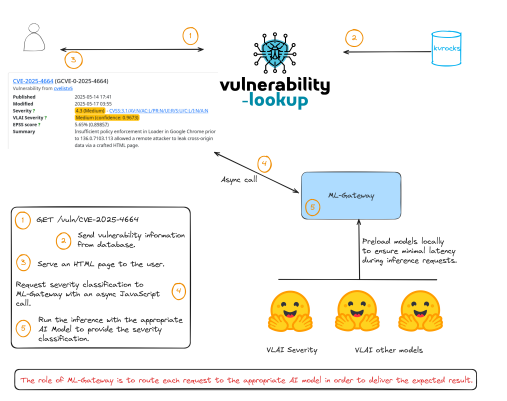
Completing Missing Data with AI



- Some vulnerabilities are published without an assigned CVSS score.
- To address this, we developed **VLAI Severity**[a], a model trained on the Vulnerability-Lookup dataset.
- **Predicts severity from the vulnerability description** before an official score is available.
- Available as a standalone model or via the CIRCL public instance.

[a] https://www.vulnerability-lookup.org/user-manual/ai/
https://www.arxiv.org/abs/2507.03607

ML-Gateway, clean RESTful API for fast inference



- **FastAPI micro-service that loads pre-trained NLP models** at start-up
- Provides per-model HTTP endpoints (`POST /classify/severity`) with OpenAPI docs
- Less than 100 ms mean latency on commodity CPU; no GPU or database required, ideal for container deployment
- Consumed asynchronously by Vulnerability-Lookup (VLAI) through the internal route
- Source code & docs[a]

[a] `https://github.com/vulnerability-lookup/ML-Gateway`

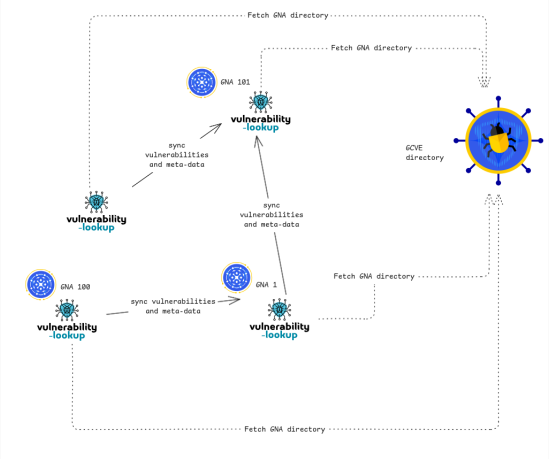
Lookup is Cool, but Publishing is Even Cooler

GCVE.eu - Role

- The primary role of GCVE[13] is to provide **globally unique identifiers** to GCVE Numbering Authorities (GNAs).
- **GNAs operate autonomously**, with full control over how they assign and manage identifiers.
- **GCVE publishes Best Current Practices (BCPs)** on directory management, Coordinated Vulnerability Disclosure (CVD), and publication protocols.
- GCVE maintains and publishes the **official directory of all GNAs**, including their publication endpoints.

[13] https://gcve.eu/

Decentralized Publication Standard



TLP:CLEAR

Core API

API Usage

- Backward compatible with `cve-search`[14]
- **Fully documented via OpenAPI**[15] — paginated, with JSON-Schema-based data validation
- The UI and core features of Vulnerability-Lookup are built entirely on top of the API
- Sighting tools[16] and other satellite projects leverage the API
- Integrated into Vulnogram (bundled with Vulnerability-Lookup) – a user-friendly interface for managing security advisories
- Supports inter-instance synchronisation (work in progress)
- Implements the MISP taxonomy[17] for objects such as comments

[14] Originally developed in late 2012
[15] https://vulnerability.circl.lu/api/
[16] https://www.vulnerability-lookup.org/user-manual/sightings/
[17] https://www.misp-project.org/taxonomies.html#_vulnerability_3

Vulnogram

Vulnogram — Mozilla Firefox

Vulnogram

https://vulnerability.circl.lu/user/submit#editor

Save | Recent vulnerabilities | My profile

NEW | Open | Download | Post to CVE.org | CVE-yyyy-nnnn.. | Load

Editor 7 | Source | Preview | CVE Portal

Vuln ID * GCVE-1-yyyy-nnnn or pick from existing — Enter GCVE-1-yyyy-nnnn format.

CVE ID CVE-yyyy-nnnn or pick from existing — cve.org

Title e.g., Memory leak in Linux Filesystem

Public at mm / dd / yyyy , -- : -- --

Problem types

e.g., CWE-20 Improper Input Validation

+ Problem type

Impacts

e.g., CAPEC-130 Excessive Allocation

+ Impact

Affected products *

Enter a vendor and product OR a package and a collection

Vendor or project — e.g., Linux

Product name — e.g., Linux Kernel

Platforms — e.g., x86, Android, Windows, MacOS, ...

Package collection URL — e.g., https://wordpress.org/plugins

Package name — e.g., kernel

Source repository (OSS) — e.g., https://git.kernel.org

Modules, components, or features — e.g., filesystem

Source-code file (OSS) — e.g., hello.c

Program routines (OSS)

+ Program routine

Versions (exact versions or ranges)

Affected?	Version (or start of a range)	< Version (range)	<= Version (range)	status changes (patches, split ranges)	versionType
y n ?	e.g., 1.2.0; 0 means no lower limits	e.g., 1.2.8, 1.2.*	e.g., 1.2.7, 1.2.*	+ item	e.g. semver, mave

+ Version

Default status (for versions not specified above) y n ?

+ Product

CVE Description | Auto Generate *

TLP:CLEAR

API Access

- **Unauthenticated, read-only** access for core look-ups (vulnerabilities, comments, bundles, sightings, . . .)
- **Registered users** can add or edit bundles, comments, and sightings
- Users with the *Commenter (Moderated)* role may edit their own bundles, comments, and sightings
- **Reporters** may edit vulnerabilities they have contributed (local sources)
- **Administrators** have full write access to all objects

Vulnerability Endpoint

- **Query vulnerabilities from all available sources** and filter by date, vendor, product, source, or CPE; results are paginated.
- Extend the response with **optional flags**:
  - `with_meta` – include extended metadata such as local updates and supplementary details
  - `with_linked` – include correlated records from other related sources
  - `with_comments` – embed user comments linked to the vulnerability
  - `with_bundles` – embed bundles referencing the vulnerability
  - `with_sightings` – embed sightings associated with the vulnerability

Related vulnerabilities

```
$ curl --silent 'https://vulnerability.circl.lu/api/vulnerability/CVE-2015-2051?with_linked=true' | jq 'keys'
[
  "containers",
  "cveMetadata",
  "dataType",
  "dataVersion",
  "vulnerability-lookup:linked"
]
```

Correlations are derived from the various sources

```
$ curl --silent 'https://vulnerability.circl.lu/api/vulnerability/CVE-2015-2051?with_linked=true' \
  | jq '.["vulnerability-lookup:linked"] | keys'
[
  "fkie_nvd",
  "github",
  "gsd",
  "variot"
]
```

You can easily get related vulnerabilities from a specific source

```
$ curl --silent 'https://vulnerability.circl.lu/api/vulnerability/CVE-2015-2051?with_linked=true' \
  | jq '.["vulnerability-lookup:linked"]["github"]'
[
  [
    "ghsa-x629-5xff-w7qg",
    {
      "schema_version": "1.4.0",
      "id": "GHSA-x629-5xff-w7qg",
      "modified": "2025-01-06T15:30:58Z",
      "published": "2022-05-17T03:11:58Z",
      "aliases": ["CVE-2015-2051"],
      "details": "The D-Link DIR-645 Wired/Wireless Router Rev. Ax with firmware 1.04b12 and...",
      "severity": [
        {
          "score": "CVSS:3.1/AV:A/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H"
        }
      ]
      <...snip...>
    }
  ]
]
```

Retrieving vulnerability sightings

```
$ curl --silent 'https://vulnerability.circl.lu/api/vulnerability/CVE-2024-5261?with_sightings=true' \
  | jq '.["vulnerability-lookup:sightings"]'
[
  {
    "uuid": "eec2c8fd-f664-4e73-b3f5-651db5fa4f3f",
    "vulnerability_lookup_origin": "1a89b78e-f703-45f3-bb86-59eb712668bd",
    "author": "9f56dd64-161d-43a6-b9c3-555944290a09",
    "vulnerability": "cve-2024-5261",
    "type": "seen",
    "source": "https://mastodon.social/users/bagder/statuses/113984646246260950",
    "creation_timestamp": "2025-02-11T09:54:37.066650Z"
  },
  <...snip...>
]
```

Pivoting with sightings

```
$ curl --silent 'https://vulnerability.circl.lu/api/sighting/?source=https://daniel.haxx.se/blog/2025/02/11/disabling-cert-checks-we-have-not-learned-much/' \
  | jq '.data[].vulnerability'
"GHSA-fq29-72jg-5hrj"
"CVE-2024-32928"
"GHSA-9mgx-552f-59p6"
"CVE-2024-56521"
"GHSA-crg3-fjm2-xvpq"
"CVE-2024-5261"
```

Additional API Endpoints

- The service provides a rich collection of endpoint families beyond `/vulnerability/`:[18]
  - **Reference data**: `/browse/`, `/capec/`, `/cwe/`, `/cisa_kev/`, `/emb3d/`, `/epss/`
  - **Collaboration**: `/bundle/`, `/comment/`, `/sighting/`
  - **Statistics**: `/stats/vulnerability/{most_commented, most_sighted}`
  - **GCVE registry**: `/gcve/registry`, integrity verification and pulls
  - **Administration & health**: `/system/`, `/user/`, `/organization/`, `/product/`
- All endpoints share consistent design principles: pagination, JSON-Schema validation, optional field masking, and comprehensive OpenAPI documentation[19]

[18] See the full specification at `https://vulnerability.circl.lu/api/`
[19] `https://www.vulnerability-lookup.org/documentation/api-v1.html`

Endpoints for Statistics

Endpoints for statistics

- Dashboard at https://vulnerability.circl.lu
- https://vulnerability.circl.lu/stats/vulnerability/most_sighted
- https://vulnerability.circl.lu/stats/vulnerability/most_commented

Generating a PDF report of the most sighted vulnerabilities:

```
$ curl -s 'https://vulnerability.circl.lu/api/stats/vulnerability/most_sighted?date_from=2025-01-01&output=markdown' \
  | pandoc --from=markdown --to=pdf -o semestrial-report.pdf
$ xdg-open semestrial-report.pdf
```

CVD - Coordinated Vulnerability Disclosure

What is CIRCL's CVD Policy?

- CIRCL has operated a Coordinated Vulnerability Disclosure (CVD) policy for over 10 years.
- A revised CVD policy[20] was introduced to align with Article 12 of the NIS 2 Directive (transposed Article 9).
- The policy governs the handling of vulnerability reports affecting ICT products (software or hardware), services, or procedural implementation.

[20] https://www.circl.lu/pub/coordinated-vulnerability-disclosure/

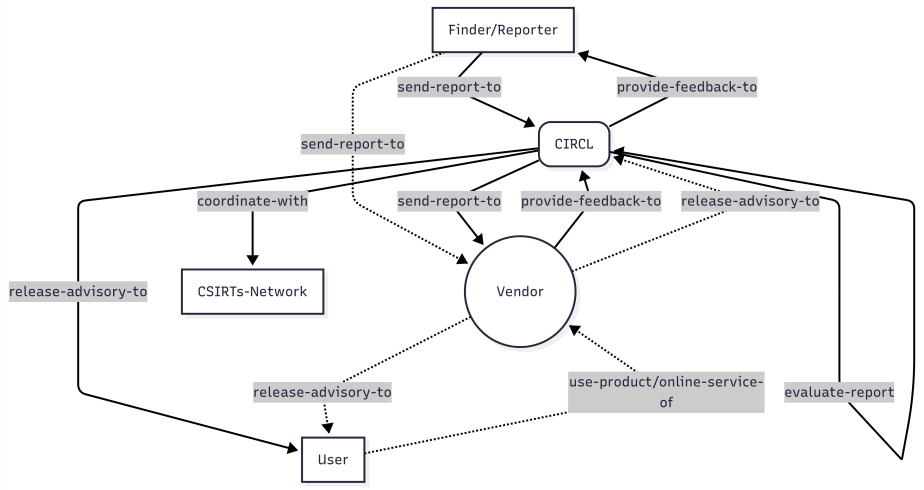
Overview - Role of CIRCL in CVD

- Acts as a trusted intermediary between the reporter and affected vendor.
- Helps to coordinate responsible disclosure while protecting security interests.

CVD Objectives

- Identify and contact the concerned vendor.
- Assist the vulnerability reporter.
- Negotiate a disclosure timeline.
- Ensure diligent follow-up measures are taken by the vendor concerned.
- Notify potentially affected users (national, EU and International).
- Contribute to ENISA's vulnerability database.

CVD Process Overview



TLP:CLEAR

Step 1: Reporting a Vulnerability

- Submit reports via the CIRCL vulnerability reporting platform: `vulnerability.circl.lu`
- Anonymity is supported and optional.
- Include sufficient technical details to enable verification and triage.
- Additional contextual information is welcome to help create effective vulnerability notifications.

Step 2: Coordination with the Concerned Entity

- CIRCL validates the vulnerability report and attempts to identify the appropriate point of contact within the affected organization.
- The validated report is then forwarded to the concerned vendor.
- CIRCL facilitates communication to ensure all parties understand the vulnerability and its potential impact.
- All interactions with the reporter are managed through the `vulnerability-lookup` platform.

Step 3: Disclosure by the Entity

- After remediation, the affected vendor is encouraged to:
  - Publish the vulnerability.
  - Provide mitigation or patch information.
- CIRCL can assist in writing or publishing the disclosure or complementary information.

Step 4: Disclosure by CIRCL or Reporter

- If the vendor:
  - Fails to respond, or
  - Does not act within the agreed timeline,
- CIRCL or the original reporter may:
  - Disclose the vulnerability publicly.
  - Inform affected users and stakeholders.

Default Remediation Timeline

- Default period: **60 days after initial notification**.
- Possible extensions based on:
  - Complexity of remediation.
  - Breadth of deployment.
  - Risk to users.

Vulnerability Disclosure Platform - User Registration

dawid-czarnecki — Mozilla Firefox

dawid-czarnecki

https://vulnerability.circl.lu/user/dawid-czarnecki

vulnerability lookup

Search Recent Admin Profile

dawid-czarnecki

Top 3 contributors Vulnerability reporter

Member since February 6, 2025.
Poland
ZigrinSecurity
2 contributions
1 vulnerability disclosure

https://zigrin.com

https://github.com/dawid-czarnecki

Recent comments

More details related to Ivanti Connect Secure stack-based buffer overflow
2 days ago

More details related to the vulnerability can be found in the CIRCL report....

CVE-2025-22457

All comments.

Recent bundles

No bundle.

All bundles.

Computer Incident Response Center Luxembourg (CIRCL)

Dumps Contributors Documentation API About

TLP:CLEAR

Vulnerability Disclosure Platform - Reporting Vulnerability 1/2

Create a vulnerability disclosure — Mozilla Firefox Private Browsing

Create a vulnerability disc

Private browsing

https://vulnerability.circl.lu/vulnerability_disclosure/new/

vulnerability lookup

Search Recent Profile

Create a vulnerability disclosure

CIRCL - Coordinated Vulnerability Disclosure (CVD) Policy

Please review the Coordinated Vulnerability Disclosure Policy before submitting a vulnerability report.

Your legal status

ZigrinSecurity

Are you reporting this vulnerability as a natural personal or a legal person.

Title

Stored XSS in Vulnerability-Lookup 2.7.0

Description Provide a detailed description of the vulnerability. Consider including how it was found, how to reproduce it, its impact, any patches or workarounds, and references.

B I H

Hi,

I discovered Stored XSS in the Vulnerability-Lookup 2.7.0.

The injection place is in Bio of a user's profile
(http://URL/user/profile). Example payload:

```
Hello world<script>alert(document.domain);</script>
```

The JavaScript code is then rendered at the user's profile view page.
For my username *regular* it's /user/regular.

I quickly managed to write a payload to get acce
whoever visits the profile. It alerts the API Key.

TLP:CLEAR

Vulnerability Disclosure Platform - Reporting Vulnerability 2/2

Edit the vulnerability disclosure — Mozilla Firefox

Edit the vulnerability discl

https://vulnerability.circl.lu/admin/vulnerability_disclosure/edit/8

Chief Executive Officer

Zigrin Security

lines: 28 words: 92 25:1

Vulnerability ID If available, provide the associated vulnerability ID (e.g., CVE identifier).

Online Service Indicates whether the vulnerability affects an online service.

Affected Products

circl:vulnerability-lookup

Provide known information about the affected product(s), including vendor, product name, and version.

Related CWEs

Optionally select any relevant Common Weakness Enumerations (CWEs) related to this vulnerability.

Related CAPECs

Cross-Site Scripting (XSS)

Optionally select any relevant Common Attack Pattern Enumeration and Classification (CAPECs) related to this vulnerability.

Remain Anonymous Check this box if the reporter does not wish to be publicly credited.

Credits

Dawid Czarnecki

Who should be credited (if not anonymous).

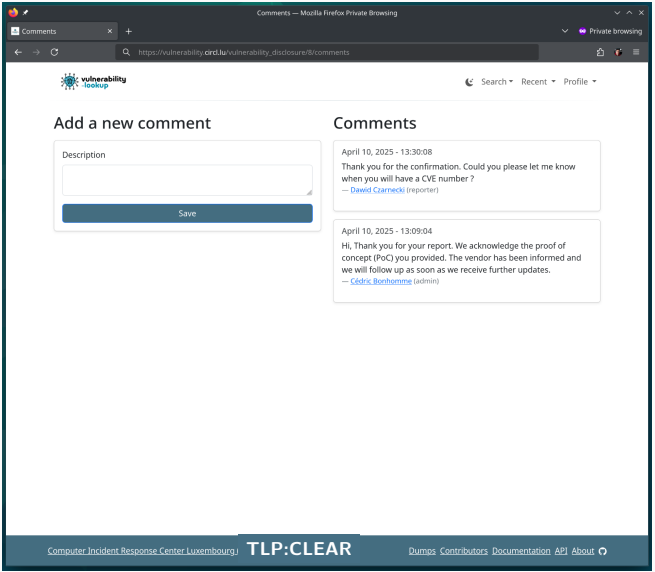
Submit

Computer Incident Response Center Luxembourg (CIRCL)

Dumps Contributors Documentation API About

TLP:CLEAR

Vulnerability Disclosure Platform - Reporting Interaction



Vulnerability Disclosure Platform - Report Management

Comments moderation — Mozilla Firefox

Comments moderation

https://vulnerability.circl.lu/admin/vulnerability_disclosures/

vulnerability lookup

Search Recent Admin Profile

Vulnerability disclosures

Search term Submit

Title	Reporter	Vulnerability	Created at	Updated at	State	Disclosure	Action
Stored XSS in Vulnerability-Lookup 2.7.0	dawid-czarnecki		2025-04-10 13:19	2025-04-10 13:24	under_review	59 days	
Vulnerability in software....	alice		2025-04-10 10:38	2025-04-10 13:23	waiting_information_from_reporter	54 days	

displaying **1** - **2** vulnerability_disclosures in total **2**

Computer Incident Response Center Luxembourg (CIRCL)

Dumps Contributors Documentation API About

Vulnerability Disclosure Platform - Disclosure Information

Vulnerability Disclosure — Mozilla Firefox Private Browsing

Create a vulnerability disc | Vulnerability Disclosure | Private browsing

https://vulnerability.circl.lu/vulnerability_disclosure/disclosure/e6377601-86da-47c5-a553-7d59053144e1

vulnerability lookup

Search Recent Profile

Disclosure Date

Time remaining for disclosure: 59 days

Disclosure date: June 09, 2025

Additional Information

Please review the disclosure deadline and ensure any necessary actions are taken.

Disclosure state: reported

Last update: April 10, 2025

Computer Incident Response Center Luxembourg | TLP:CLEAR | Dumps Contributors Documentation API About

Vulnerability Disclosure Platform - CVE Creation

• CVE-2025-32413 — Mozilla Firefox

• CVE-2025-32413

https://vulnerability.circl.lu/user/edit/cve-2025-32413#editor

Save | Vulnerability page | Delete | You are editing cve-2025-32413 from the 'cvelistv5' source. Changes made to this vulnerability will be commited in the local source with a different id. | Recent vulnerabilities | My profile

NEW | Open | Download | Post to CVE.org | CVE-yyyy-nnnn.. | Load

Edited

Editor 12 | Source | Preview | CVE Portal

Vuln ID * CIRCL-SA-2025-32413

CVE ID CVE-2025-32413 cve.org

Title Vulnerability-Lookup before 2.7.1 allows stored XSS

Public at mm / dd / yyyy , -- : -- --

Problem types

CWE-79 Improper Neutralization of Input During Web Page Generation (XSS or 'Cross-site Scriptir

+ Problem type

Impacts

+ Impact

Affected products *

Vendor or project: CIRCL

Product name: Vulnerability-Lookup

Platforms: eg., x86, Android, Windows, MacOS, ..

Package collection URL: eg., https://wordpress.org/plugins

Package name: eg., kernel

Source repository (OSS): eg., https://git.kernel.org

Modules, components, or features: eg., filesystem

Source-code file (OSS): eg., hello.c

Program routines (OSS): + Program routine

Versions (exact versions or ranges)

Affected?	Version (or start of a range)	< Version (range)	<= Version (range)	status changes (patches, split ranges)	versionType
y n ?	0	2.7.1	eg., 1.2.7, 1.2.*	+ item	semver

+ Version

Default status (for versions not specified above) y n ?

+ Product

CVE Description | Auto Generate

TLP:CLEAR

Closing

Future Development

- Deeper analysis of the content and context of sightings, including **source reliability assessment**.
- Full-text search capabilities across all integrated sources.
- Integration of scoring models such as Vuln4Cast[21], with testing planned on our dataset to enhance reproducibility.
- **Improved notification capabilities** for newly observed vulnerabilities via webhooks.

> The project is evolving rapidly — feedback and feature suggestions are always welcome!

[21] https://github.com/FIRSTdotorg/Vuln4Cast

References

https://www.vulnerability-lookup.org

CIRCL public instance https://vulnerability.circl.lu

Source code https://github.com/vulnerability-lookup/vulnerability-lookup

Dataset, AI Model Training, Models
https://github.com/vulnerability-lookup/VulnTrain