Scoring vulnerabilities by leveraging activity data from the Fediverse

2025 Cyber Threat Intelligence Conference
Berlin

Cédric Bonhomme
Computer Incident Response Center Luxembourg
cedric.bonhomme@circl.lu [57B7 A70D]

Alexandre Dulaunoy
Computer Incident Response Center Luxembourg
alexandre.dulaunoy@circl.lu [44e6 cbcd]

Abstract

In this short paper, we explore the topic of vulnerability observations using diverse sources, including distributed networks like the Fediverse, general websites (such as RSS/Atom feeds from technical sites), and the AIL[1] project. Given that Vulnerability-Lookup is a new project, the first section will provide essential context. The core will be presented in the second section. The tool and its capabilities will be showcased in detail during the presentation.

1 Introduction

Vulnerability Lookup[2] facilitates quick correlation of vulnerabilities from various sources, independent of vulnerability IDs, and streamlines the management of Coordinated Vulnerability Disclosure (CVD). It is a rewritten and enhanced version of cve-search, an open-source tool originally designed to maintain a local CVE database.

1.1 Why this new vulnerability database project ?

For over a decade, cve-search has been maintained and operated by CIRCL. However, we identified design and **scalability limitations** in the original implementation, with its public instance currently maxing out at 20,000 queries per second. Another aspect is the **diversification of vulnerability sources** beyond just the NVD CVE. The new tool now supports additional sources such as GitHub, PySec, GSD, and CSAF. This broadening of sources aligns with our commitment to support the CVD process, and extending vulnerability information in a collaborative manner. Vulnerability Lookup address these two issues: it is fast, versatile and ease the editing of vulnerability advisories. Our aim is to go beyond and implement innovative features tailored for CVE Numbering Authorities (CNAs), cybersecurity vendors, incident response teams, vulnerability reporters, and developers. Vulnerability Lookup is released under the AGPLv3 license. The main instance is operated by CIRCL[3].

[1] https://ail-project.org
[2] https://github.com/cve-search/vulnerability-lookup
[3] https://vulnerability.circl.lu

1.2 Main functionalities

Vulnerability Lookup is accessible via a Web interface and an extensive HTTP API with a Swagger documentation. It provides an interface to search publicly known information from security vulnerabilities in software and hardware along with their corresponding exposures. Vulnerability Lookup currently features:

- an intuitive graphical user interface. For creating and editing CVE information, we use the Vulnogram project[4], which integrates perfectly with our tool.
- a modular system for importing from various vulnerability sources.
- a fast lookup API to search for vulnerabilities and find correlations.
- users can comment on security advisories and create/share bundles.
- an extensive RSS and Atom.
- sightings.

We recently introduced the capability to leverage the MISP's taxonomy for vulnerabilities[5]. Tags from the taxonomy are stored automatically in the meta field of the objects. This paves the way for leveraging more taxonomies in the future. In the latest release we have integrated the Japan Database of Vulnerability Countermeasure Information (JVN DB), correlating security advisories from the multiple sources already available in Vulnerability Lookup.

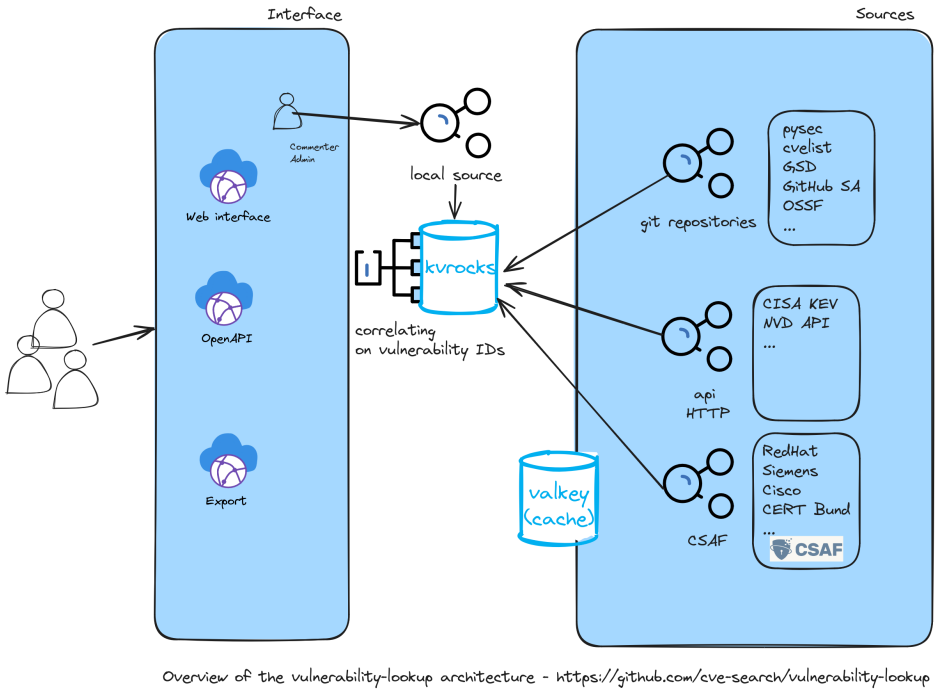

Figure 1: Vulnerability-Lookup high level architecture

This architecture will be detailed during the presentation.

[4] https://github.com/Vulnogram/Vulnogram
[5] https://www.misp-project.org/taxonomies.html#_vulnerability_3

2 Sightings

Sightings are essentially observations of specific vulnerabilities. They provide tangible evidence that a vulnerability is not just theoretically present but actively observed in the wild, either in an organization's environment or through detected exploitation attempts by threat actors.

By tracking sightings, organizations can prioritize vulnerability remediation based on actual occurrences within their own environment or industry. A sighting indicates an immediate need to address the vulnerability to mitigate risk.

Type of sightings:

Type	Description	Negative/Opposite
seen	The vulnerability was mentioned, discussed, or seen somewhere by the user.	-
confirmed	The vulnerability is confirmed from an analyst perspective.	X
exploited	This vulnerability was exploited and seen by the user reporting the sighting.	X
patched	This vulnerability was successfully patched by the user reporting the sighting.	X

Table 1: Types of vulnerability sightings

Key aspects of sightings:

- the **source** of the sighting: report, blog post, status (Fediverse or Twitter);
- the **type** of the sighting (see the previous table);
- the **vulnerability**: any vulnerability identifiant.

Example of a sighting in Vulnerability-Lookup:


```
{
    "uuid": "f6ed692b-2656-4ce0-bcf1-eaf12dfe281d",
    "vulnerability_lookup_origin": "1a89b78e-f703-45f3-bb86-59eb712668bd",
    "author": "8dfa6142-8c6d-4072-953e-71c85404aefb",
    "type": "seen",
    "source": "https://infosec.exchange/users/cve/statuses/113389560858828548",
    "vulnerability": "CVE-2024-10312",
    "creation_timestamp": "2024-10-29T08:36:31.492184Z"
}
```


Twitter's restriction on free API access has impacted many communities that relied on its data. Meanwhile, Twitter is increasingly being abandoned in favor of Mastodon, especially within the infosec community. As a result, other projects are beginning to turn to sources like Mastodon, where activity on this distributed network is rapidly growing. This shift is why the Fediverse is our primary source for sightings. However, we plan to further leverage other sources.

In Vulnerability-Lookup, a sighting can be recorded either by an expert or programmatically through various tools that can easily integrate with its API. FediVuln[6] is capable of connecting to the activity stream of any ActivityPub-based service, such as Mastodon, Lemmy, and similar platforms. Using regular expressions, it detects various types of vulnerabilities (e.g., CVE, GHSA, PySec, CSAF CERT-Bund, CSAF RedHat, CSAF CISCO, and GSD) within status content. The filtered statuses can then be used to automatically create Sighting objects.

[6] https://github.com/CIRCL/FediVuln

2.1 Correlating EPSS score with Sightings

Since the beginning of the project we are interested in EPSS and Vuln4Cast[7]. We recently integrated EPSS into Vulnerability-Lookup, and are aiming to gain a deeper understanding of its model. We would like to compare its predictions with our own findings.

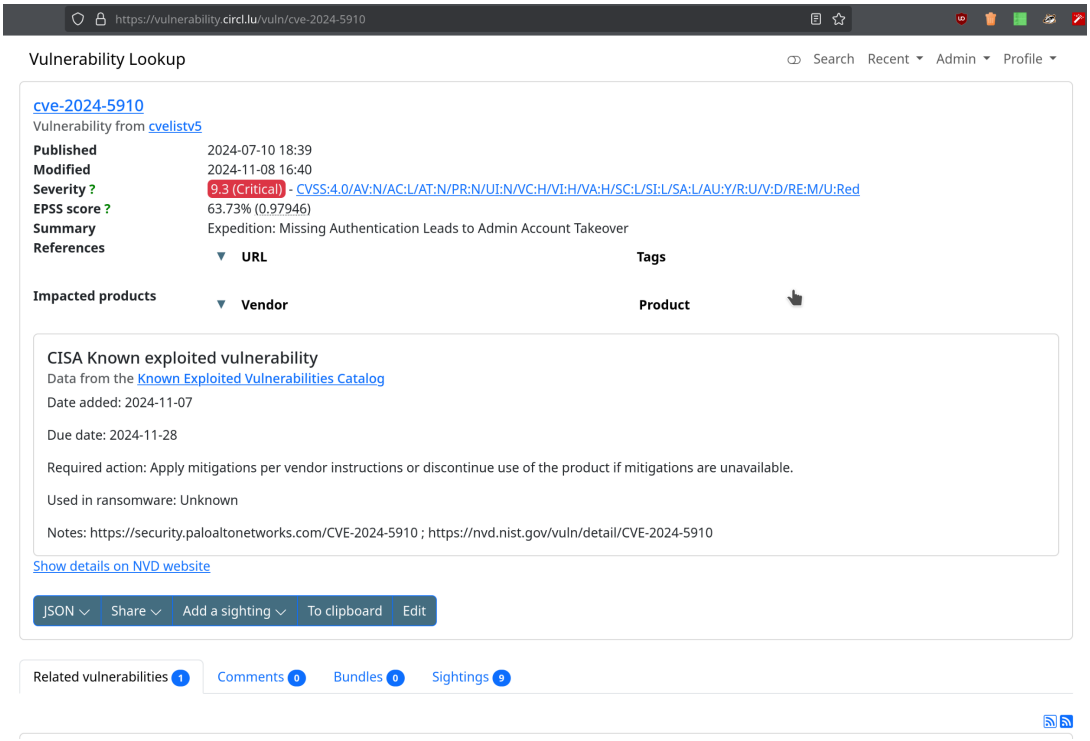

Figure 2: EPSS score for a specific vulnerability.

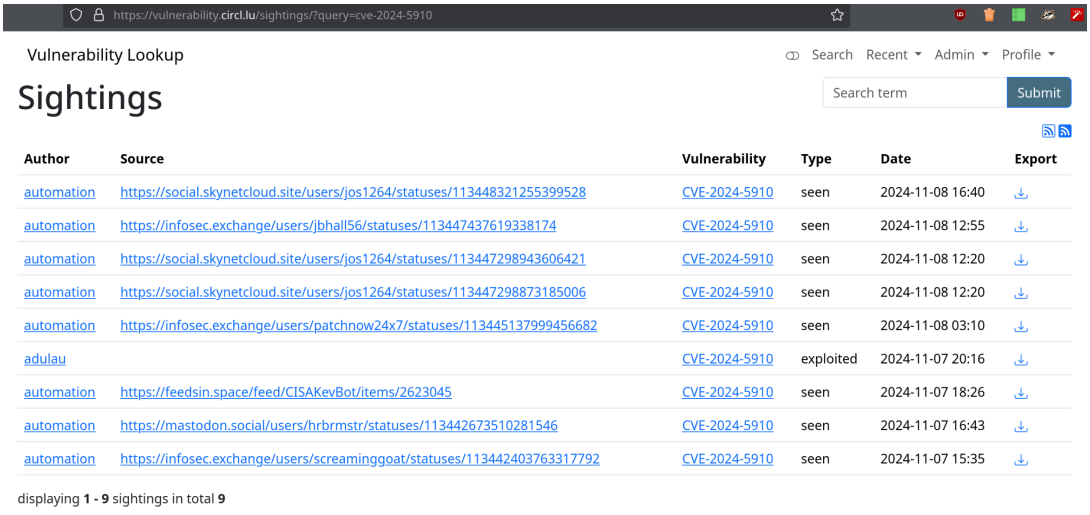

Figure 3: Sightings collected from the Fediverse for the same vulnerability.

In this instance, the EPSS prediction appears to align with our sightings, which were retrieved in only 24 hours. We encountered several similar examples.

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

2.2 Tracking the exploitability of vulnerabilities prior to their public disclosure

We observed that vulnerabilities are frequently discussed across the web before they are officially published, sometimes from a few hours up to several weeks in advance. For this reason, it is important for us to track activity related to these vulnerabilities and to provide the community with tools to respond effectively. To address this need, we have enabled features for commenting, bundling, and reporting sightings of unpublished vulnerabilities when relevant observations are found.

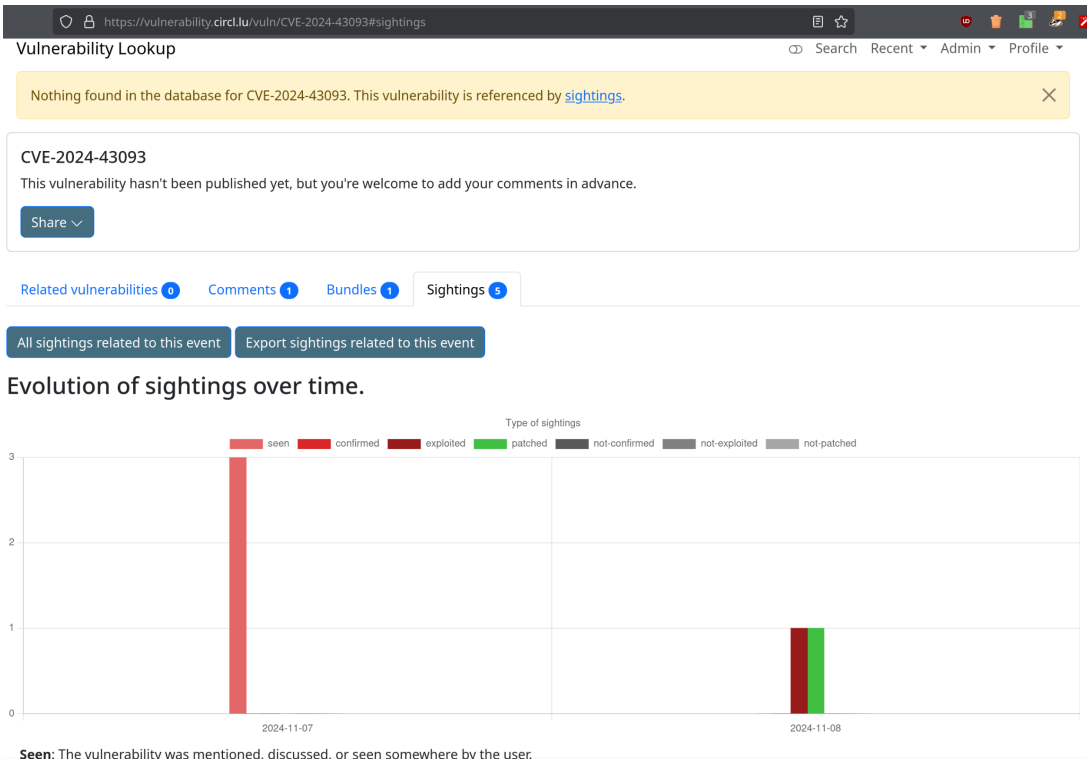

Figure 4: Sightings and comments on a vulnerability not published.

We believe that enriching vulnerability information before its public release can be highly beneficial for analysts. When people actively seek or exchange information about a vulnerability, it signals that the issue should be prioritized. We aim to provide the necessary tools to support this effort.

3 Conclusion

Using data from the Fediverse to generate sightings is an initial step. As mentioned earlier, we also plan to incorporate additional sources, including MISP and the AIL framework, to enhance the comprehensiveness of our observations[8].

We are also considering the use of more advanced regular expressions (or combinations), source reputation (scoring), and LLM-based techniques to enhance the automatic detection. Indeed, by conducting a deeper analysis of the content, it becomes possible to automatically classify sightings. Currently, all sightings are simply labeled as “seen” but these improvements would enable a better categorization (“patched”, “exploited”, etc.).

[8]With AIL, we will be able to gather information from sources like Pastebin, Gist, and others.