Zero-Day Vulnerabilities, Cartography, and Quantification Initiative

Myriam Ouraou
Thales Digital Factory, Paris, France
De Vinci Higher Education, De Vinci Research Center, Paris, France

myriam.ouraou@thalesgroup.com

Abstract: Zero-day vulnerabilities have become a critical concern across security teams. Despite their growing frequency, they remain a largely uncharted domain: there is currently no consolidated information, statistical visibility, or cartography describing the true scale of the problem. Organisations lack metrics that would enable them to understand how zero-days are distributed among products or technological ecosystems, limiting their ability to anticipate, detect, and accurately assess the risks associated with these vulnerabilities. In the absence of a structural understanding of the phenomenon, organisations rely on linear, product-centred approaches inherited from traditional vulnerability management and built around indicators such as CVE identifiers, CVSS scores, or patch availability. Although these indicators are essential, they do not capture the multidimensional relationships and ecosystem-level dependencies that shape zero-day behaviour. Consequently, analyses remain confined to individual vulnerabilities rather than the structures that connect them, leaving organisations without models capable of describing how zero-days emerge, cluster, or propagate across technologies. To address this gap, this work begins with an in-depth *a posteriori* phase aimed at establishing the first consolidated inventory of past zero-days. This fundamental effort is intended to establish an initial structural basis that will enable further long-term research into the systemic characteristics of zero-day vulnerabilities. Afterwards, this study seeks to reveal "domino effects", clusters of technological weakness, and cross-product propagation paths that remain invisible in traditional analytical frameworks. The goal is to provide a new visualisation to support not only future predictive models and improved detection strategies, but also more informed and contextualised risk assessments. By moving beyond linear scoring systems and embracing a structural, graph-driven perspective, this work lays the groundwork for a more proactive, comprehensive understanding of zero-day vulnerabilities and the technological ecosystems they inhabit.

Keywords: Vulnerability, Zero-day, Cartography, Graph

1. Introduction

A zero-day is a vulnerability that has not been publicly disclosed and for which there is no known fix. The existence of such a flaw in a product means that there is no protection, either temporary or permanent. They are more dangerous than traditional vulnerabilities because they emerge in a space of total asymmetry: no patch or mitigation measure exists at the time of their discovery, leaving defences structurally behind. In a context where disclosure and remediation cycles generally provide a framework for managing known vulnerabilities, zero-days impose a total break, especially when their number keeps increasing (Figure 1). They reveal not only the absence of immediate protection but also the limitations of our analysis models, which are designed to react to existing threats rather than anticipate the mechanisms that promote the emergence of these invisible vulnerabilities. The potential damage from them can be substantial and their challenge lies in our limited understanding of them. We still do not have consolidated metrics, models, or frameworks that tell us how zero-days behave as a group. We know how to handle a vulnerability once it becomes public, but we do not fully understand the conditions that lead to its emergence. This brings the central problem of this paper into focus: For years, our industry has depended on the same indicators, CVE identifiers (Common Vulnerabilities and Exposures), CVSS scores (Common Vulnerability Scoring System), and patch availability. These indicators are vital, but they are inherently reactive. They show us what has already happened, not what is happening structurally beneath. When we talk about zero-days, the problem becomes even clearer: a zero-day is, by definition, discovered too late. Rather than asking ourselves how we can correct them more quickly, we should ask ourselves whether we could anticipate their future occurrence through better understanding. We used to analyse zero-days as a singularity, but we never established a correlation between the different conditions that lead to their discovery and exploitation. We may wonder whether there are similar patterns, common mechanisms for their emergence, that we can model and highlight. The idea is not to predict the exact next zero-day but to identify where it is more likely to appear and how. By analysing connections between vulnerabilities, shared components, recurring exploit primitives, or historical sequences of related weaknesses, we can start building a structural understanding that does not exist today. This structural understanding, in turn, can guide organisations in prioritising monitoring, testing, and hardening efforts, not based only on CVSS, but based on the actual dynamics of their vulnerability ecosystem.

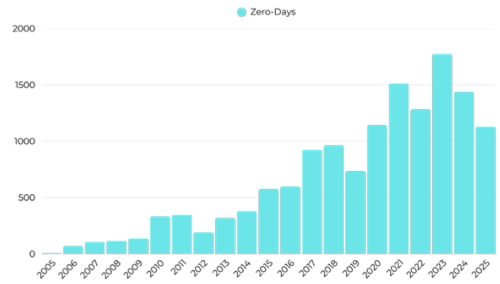

Figure 1: Number of zero-day vulnerabilities per year (Zero Day Initiative)

2. Related Work

Empirical studies on zero-day vulnerabilities have mainly focused on measuring the lifecycle and dynamics of patches. Bilge and Dumitras conducted the first large-scale field study on zero-day attacks, identifying 18 vulnerabilities exploited before public disclosure on 11 million real hosts and showing that a typical zero-day vulnerability persists for an average of 312 days. Ablon and Bogart characterised the zero-day ecosystem in more detail from a strategic perspective, estimating that approximately 5.7% of stockpiled vulnerabilities are independently discovered by others within a year. More recently, Roumani applied survival analysis to 4,897 ZDI vulnerabilities to model patch release times, while Shet and Alsmadi analysed ZDI disclosures from 2024 using a machine learning-based severity classification. Although this work provides valuable statistical insights, it describes vulnerability attributes in isolation and does not capture the relational structures that link vulnerabilities across products, weakness types, and exploitation contexts. Yin et al. introduced a compact vulnerability knowledge graph oriented toward risk assessment. However, none of these approaches are tailored to zero-day vulnerabilities specifically, nor do they incorporate community detection to reveal latent structural groupings. The present work addresses this gap by constructing a heterogeneous graph dedicated to zero-days and applying the Leiden algorithm and UMAP to uncover structural communities and visualise their organisation beyond what conventional metrics such as CVSS or EPSS can capture.

3. Dataset Creation

The primary challenge with zero-day vulnerabilities lies in the fact that we must wait for their exploitation to become aware of their existence. Consequently, we are limited to studying the past rather than predicting the future. This necessitates conducting an *a posteriori* analysis by examining their characteristics. As explained in the introduction, what makes a zero-day vulnerability particularly dangerous is that no one is aware of its existence until an exploit is deployed in real-world situations. (Figure 2).

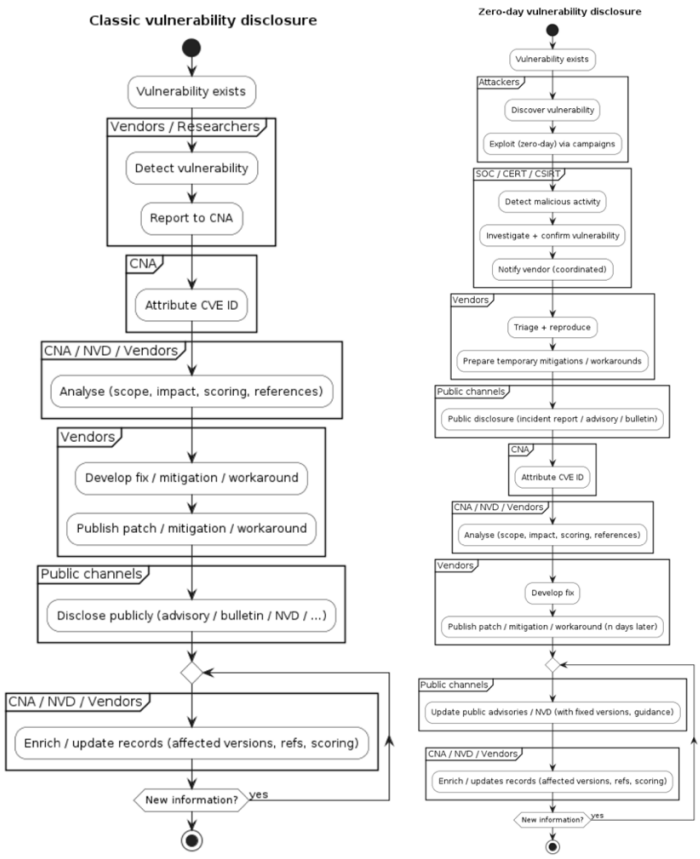

Figure 2: Vulnerability lifecycle

To better understand zero-days, we need to analyse the different patterns that make them zero-day vulnerabilities. The advantage of identifying them like other vulnerabilities is that we have access to the same data about them, all recorded under a single CVE identifier. However, since all zero-days are recorded as CVEs without any distinction, no official source, such as MITRE, NVD, CISA, etc., currently allows zero-days to be isolated, categorised, or quantified accurately. In practice, they disappear statistically "into the mass" of CVEs, which prevents us from obtaining a consolidated view of their volume, their distribution by publisher, product, or technology sector, or even their evolution over time. In the absence of formal recognition in official databases, monitoring of zero-days therefore relies, at the time of writing, on non-institutional inventories maintained by researchers, threat intelligence teams, or certain cybersecurity companies, which attempt to record every known case. These sources, although valuable, remain heterogeneous, non-standardised, and biased, which makes it even more difficult to build a comprehensive and methodologically robust understanding of the zero-day phenomenon.

This is why this initial research phase will draw on Trend Micro's Zero Day Initiative (ZDI) database, as did the authors cited above, to evaluate the results of this initial approach. This choice will enable us to analyse the relevance of the method in a controlled environment, pending the enrichment and refinement of the zero-day dataset in future work. Each zero-day vulnerability identified by ZDI was retrieved either under a CVE ID, if it was associated with one, or under a ZDI ID if it was not, in the form ZDI-YY-XXX. We enriched each with different metrics from open-source data:

- CVSS Score, which measures the severity of the vulnerability;
- CPE (Common Platform Enumeration), which specifies the products and versions concerned;
- EPSS Score (Exploit Prediction Scoring System), which measures the probability that a vulnerability will be exploited within 30 days of its publication;
- KEV affiliation (Known Exploited Vulnerabilities), catalogue that lists vulnerabilities confirmed as being actively exploited;
- CWE (Common Weakness Enumeration), which provides details about the type of vulnerability exploited;

- CAPEC (Common Attack Pattern Enumeration and Classification), which characterises the typical attack patterns and exploitation strategies associated with the vulnerability;
- ATT&CK Technique, which maps the vulnerability to concrete adversarial behaviours and operational techniques used in real-world attacks;
- Threat Actors, known to exploit specific CVE.

The generated dataset comprises 12,342 CVEs between 2005 and 2025.

4. Graph Approach

This section presents the first methodological phase of this work devoted to modelling zero-day vulnerabilities. The goal is to identify latent structures and recurring threat profiles.

4.1 Construction

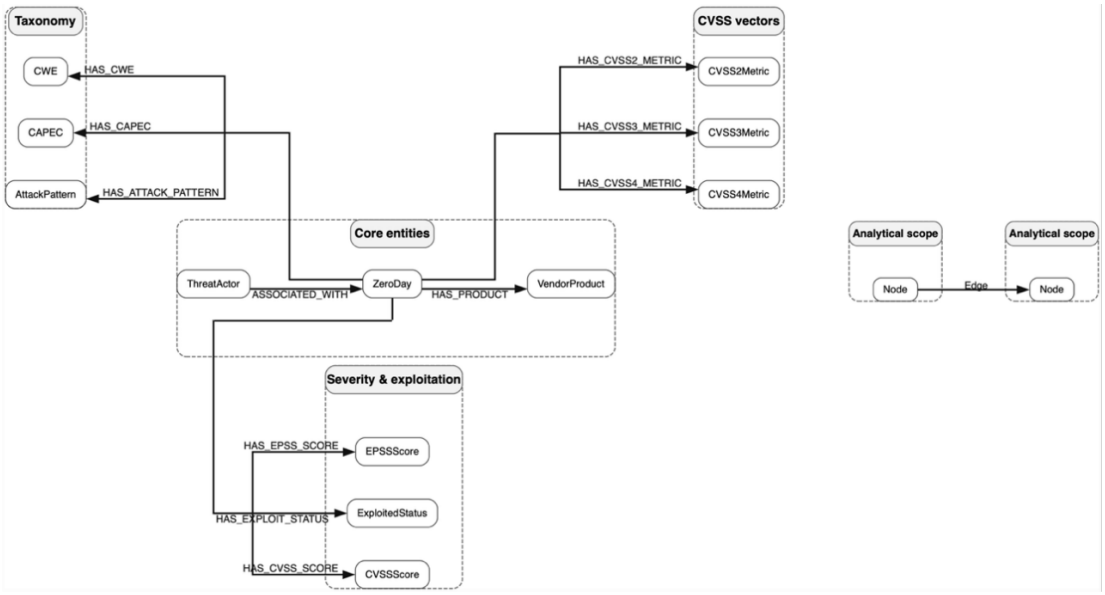

Figure 3: Pseudo-graph structure

The graph is constructed using the database generated in the previous section. It is a heterogeneous oriented graph $G = (V, E)$ where the set of nodes is partitioned into subsets. Each node $v \in V$ is associated with a type $\tau(v)$ belonging to a finite set of types and a vector of attributes. The set of edges $E \subseteq V \times V$ is also typed, with each relationship has a specific meaning, such as “affects”, “exploits”, “is associated with”, etc.

4.2 Matrix Projection

To enable algorithmic exploitation of the graph, we project the relational structure into a matrix space. The first projection is based on the construction of a direct context matrix $B \in \mathbb{R}^{|Z| \times |C|}$, where Z denotes the set of zero-day nodes and C the set of context nodes. A context node corresponds to all entities directly linked to a zero-day by a semantic edge. Formally, for a zero-day $z_i \in Z$ and a context node $c_j \in C$, the coefficient B_{ij} is 1 if and only if there is an oriented edge $(z_i, c_j) \in E$ or 0 otherwise. This matrix is deliberately binary to capture the presence or absence of relationships regardless of their frequency. The result is a matrix where each row corresponds to a zero-day and each column to an element of its context.

4.3 TF-IDF Weighting

In a second step, a weighting stage is carried out to reassess the relative importance of contexts based on their overall rarity. To do this, matrix B is transformed using TF-IDF (Term Frequency-Inverse Document Frequency) weighting, commonly used to reinforce discriminating dimensions. Here, each zero-day is equivalent to a document and each context node to a term describing that document. Since the matrix B is binary, the term frequency $\mathrm{tf}(z_i, c_j)$ is defined as 1 if $B_{ij} = 1$ and 0 otherwise. The document frequency $\mathrm{df}(c_j)$ corresponds to the number of zero-days associated with context c_j. Based on this quantity, we define a weight of importance inversely proportional to the overall frequency of the context: the more frequent a context is, the less informative it is for distinguishing zero-days. In this work, we adopt an adapted version of the weighting factor defined by:

$$\mathrm{idf}(c_j) = \log\left(\frac{1+|\,Z\,|}{1+\mathrm{df}(c_j)}\right) + 1.$$

The $+1$ in the numerator and denominator avoids degenerate cases when $\mathrm{df}(c_j) = 0$. The $+1$ added after the logarithm prevents contexts present in the zero-day set from being completely cancelled out by IDF weighting. These contexts thus retain a strictly positive weight, which guarantees a complete and stable representation of the context space. The weighted matrix $B^{\mathrm{tfidf}} \in \mathbb{R}^{|Z|\times|C|}$ is then obtained by applying the weighting factor column by column, according to:

$$B_{ij}^{\mathrm{tfidf}} = \mathrm{tf}(z_i, c_j) \cdot \mathrm{idf}(c_j).$$

To make representations comparable between zero-days regardless of their degree of connectivity, L_2 normalisation is then applied line by line. This normalisation consists of dividing each row by its Euclidean norm: for a zero-day z, if we denote x_z as the corresponding row vector in B^{tfidf}, we get $\hat{\mathrm{x}}_z = \frac{\mathrm{x}_z}{\|\mathrm{x}_z\|_2}$. This operation reduces the influence of the degree of a zero-day and emphasises the direction of the vector.

4.4 Community Detection

4.4.1 Cosine similarity

The next step is to identify vulnerability groups with similar contextual profiles by constructing a similarity graph between zero-days and applying a community detection algorithm. We start with the matrix $B^{\mathrm{tfidf}} \in \mathbb{R}^{|Z|\times|C|}$ obtained previously. From each normalised representation, we define a measure of similarity between two zero-days z_i and z_j using cosine similarity:

$$s_{ij} = \cos\left(\hat{\mathbf{x}}_i, \hat{\mathbf{x}}_j\right) = \frac{\hat{\mathbf{x}}_i \cdot \hat{\mathbf{x}}_j}{\|\,\hat{\mathbf{x}}_i\,\|_2 . \|\,\hat{\mathbf{x}}_j\,\|_2} = \hat{\mathbf{x}}_i^{\top}\hat{\mathbf{x}}_j, \quad s_{ij} \in [0;1]$$

We choose the cosine similarity measure because an Euclidean norm-based method would favour the most connected zero-days. Here, cosine similarity applied to vectors normalised in L_2 depends only on their normalised scalar product. This makes it possible to compare zero-days based on the relative composition of their context, regardless of their degree.

4.4.2 KNN

To avoid the quadratic cost of exhaustive pair-by-pair similarity calculations for all nodes in the graph, a k-nearest neighbours (kNN) graph is constructed, where each node z_i is connected to its k most similar neighbours, provided that this similarity exceeds a minimum threshold θ. θ acts as a filtering mechanism to eliminate weak similarity links that are more likely to introduce noise. This gives us a non-oriented and non-redundant graph $G_{\mathrm{sim}} = (Z, E_{\mathrm{knn}})$, whose edges are weighted by similarity s_{ij}. The hyperparameters of the similarity graph were selected empirically based on a comparative analysis of several pairs (k, θ). The aim is to limit excessive fragmentation of the graph, to ensure the existence of a dominant connected component (DCC) and to preserve sufficient contrast between zero-days. A DCC enables the propagation of local similarity information on a global scale and community detection guided by the intrinsic structure of the data rather than by induced fragmentation. Experimental results show that the configuration $k = 35$ and $\theta = 0.05$ provides the best trade-off between connectivity and discrimination. This setting produces a graph without isolated nodes, a DCC covering approximately 99% of the zero-day corpus, and a limited number of connected components overall. At the same time, the distribution of similarity weights remains sufficiently spread out, thus preserving significant contrasts between vulnerabilities. This balance ensures that the similarity graph is both globally connected and locally informative.

4.4.3 Leiden

Community detection is performed using the Leiden algorithm by partitioning the similarity graph so that the nodes that are most strongly connected to each other relative to the rest of the graph form a community. This algorithm optimises an objective function that evaluates the quality of the partitioning by comparing the observed density of intra-community links with that expected in a null model, where the graph structure is random while maintaining the degree of each node. A community will be considered relevant when it has an excess of internal connections relative to the degrees alone. The function introduces a resolution parameter γ that serves as a scale: the smaller it is, the more it will favour large communities, and vice versa. To avoid an

arbitrary choice of the resolution parameter, a sensitivity analysis was conducted by varying γ over a grid of values $\gamma \in \{0.5,0.75,1.0,1.25,1.5,2.0,2.25,2.5,2.75,3.0\}$. For each value of γ, community detection was repeated for several seeds to assess the robustness of the results with respect to the randomness of the algorithm. The choice of the γ parameter is based on several complementary criteria:

- total number of communities detected
- distribution of community sizes
- relative size of the dominant community
- proportion of very small communities
- stability of the results with respect to the choice of random seed

The results obtained are generally stable for the seed considered, both in terms of the number of communities and the distribution of sizes. To assess the robustness of the detected communities, we also measured the stability of the partitions obtained using the Adjusted Rand Index (ARI). ARI assesses the concordance between partitions obtained with different random seeds by correcting for chance: values close to 1 indicate almost identical groupings, while values close to 0 indicate concordance that is no better than chance. Consistently high ARI values (≈ 0.93) indicate that the community structure is very stable and largely independent of random initialisation. Therefore, a fixed seed (seed = 42) is used for the final partition to ensure full reproducibility of the results. A monotonic evolution of partitioning is observed as γ increases: the number of communities gradually increases, while the size of the dominant community decreases. The value $\gamma = 2.5$ was chosen because it represents the best compromise between these two extremes. This value therefore, provides a good level of interpretable and stable resolution without introducing artificial fragmentation. This leaves us with 71 communities.

4.5 Profiling

Finally, to make each community interpretable, three different types of profiling are performed.

4.5.1 Coverage

Based on the partitioning obtained by Leiden, we consider for each community k the set of zero-days that compose it, Z_k. For each context c_j defined within the binary matrix B, we calculate its number of occurrences in the community and then normalise it by the size of the community to account for its coverage within the latter:

$$\mathrm{cov}(c_j, k) = \frac{1}{|\, Z_k \,|} \sum_{z_i \in Z_k} B_{ij}.$$

Coverage therefore represents the proportion of zero-days that share the same context within the community. The higher the coverage of a context, the more it can be considered structuring. This initial profiling highlights the most common contexts within each community but does not target more specific ones.

4.5.2 TF-IDF importance

With this second profiling, we want to incorporate the notion of weighted importance for each context. We therefore use the TF-IDF matrix: within each community k, we calculate the average TF-IDF weights for each context c_j:

$$\mu_{\mathrm{tfidf}}(c_j, k) = \frac{1}{|\, Z_k \,|} \sum_{z_i \in Z_k} B_{ij}^{\mathrm{tfidf}}.$$

This quantity measures the average importance of each context within a community. The higher the average, the more prevalent it will be among several members of the community and the more relatively rare it will be on the global scale of the graph. Unlike different profiling, this type of profiling penalises overly generic contexts to highlight those that specifically characterise each community.

4.5.3 Lift

The final profiling step consists of comparing the frequency of a context within its community to its overall frequency within the corpus. To do this, we calculate the ratio between the coverage of a context within its community $\mathrm{cov}(c_j, k)$ and its overall coverage $\mathrm{COV}(c_j)$, defined by:

$$COV(c_j) = \frac{1}{N}\sum_{i=1}^{N} B_{ij}\,,$$

where N denotes the total number of zero-days. The lift (over-representation factor) is then defined as:

$$\text{lift}(c_j, k) = \frac{cov(c_j, k)}{COV(c_j)}.$$

A lift greater than 1 indicates that the context is overrepresented in the community compared to the rest of the corpus and is therefore a distinctive marker of the group. Conversely, a lift less than 1 indicates that the context is underrepresented. A lift close to 1 indicates a generic context whose presence is not specific to a given community. This profiling thus makes it possible to identify the truly discriminating attributes of each community, regardless of their absolute frequency.

5. Results

5.1 Global Community Structure of Zero-day Vulnerabilities

Overall, the communities generated are neither trivial product-based groupings nor purely taxonomic groupings. Rather, they emerge from the interaction between CWE, affected technology ecosystems, exploitability indicators, severity measures and, in a very limited number of cases, explicit associations with malicious actors. From a macroscopic perspective, the partitioning highlights a clear distinction between large cross-cutting communities and more specialised clusters. The largest communities are characterised by generic vulnerability patterns associated with a wide dispersion of affected products with characteristics similar to those identified in the previous section. In contrast, smaller communities tend to exhibit greater internal consistency, often driven by specific software ecosystems and attack patterns. The various profiling strategies applied to communities have brought to light more specific and structured information for each community, providing a different perspective on traditional coverage analysis. Communities are structured around a small number of dominant technical signals, generally a combination of vulnerability weakness and technological products, while exploitability and severity measures act as modulators rather than the main drivers of grouping. In particular, there is a strong emphasis on high CVSS contexts and low or moderate EPSS scores. Some communities show signs of exploitation by threat actors, but this context never appears to be a determining factor, but rather a secondary one. Let us introduce a few communities selected for certain characteristics:

- Community 0 (n=555): This community constitutes the largest group identified in the graph and can be interpreted as the structural backbone of the zero-day corpus. It includes a significant proportion of vulnerabilities that are not publicly observed as exploited and are mainly associated with low EPSS scores, although they have high severity levels according to CVSS v3. It is dominated by weaknesses related to memory management across a diverse range of software products and platforms. However, profiling based on lift reveals a more nuanced picture. Although no single product dominates the community, there is a clear overrepresentation of technologies related to industrial software, virtualisation environments and network infrastructure components.
- Community 13 (n=300): It forms a very coherent group focused on injection-type vulnerabilities. This is reflected in the CAPEC and ATT&CK layers, which are almost “saturated” in terms of coverage (almost 100%). From a risk perspective, the community is dominated by High or Critical vulnerabilities, while the probability of exploitation remains mainly low with a small high fraction and a limited portion exploited in the wild, suggesting selective and likely opportunistic operational exploitation. Coverage mainly highlights Ivanti and TrendMicro products and OT-oriented components. At the same time, there is a low presence of several threat actors, which shows the attackers' interest in security technologies.
- Community 51 (n=92): It corresponds to a server-side cluster and enterprise infrastructure, largely shaped by weaknesses in resource management and isolation, particularly accentuated by very high lift values. Technologically, the community relies on management and collaboration ecosystems and infrastructure and industrial components. The community is primarily classified as CVSS v3.X High/Critical and strongly aligned with remotely exploitable conditions, while still associated with low to moderate EPSS classes and only a small fraction is observed to be exploited in the wild. Finally, although several malicious actor nodes appear in the coverage and lift, their individual support remains low and none dominate, suggesting opportunistic behaviour.

The global analysis highlights dominant contexts characterised by low or moderate EPSS scores and a majority of vulnerabilities not publicly observed as exploited. Nevertheless, these contexts constrain us in the evaluation of structural patterns of zero-days known to be exploited. To better identify them, we apply the same methodology to a reduced corpus composed exclusively of zero-day vulnerabilities known to be exploited in nature, as referenced in the CISA KEV catalogue. This filtering step significantly reduces the corpus size from 12,342 to 678 vulnerabilities and leads to 23 communities. We choose here $k = 15$, $\theta = 0.05$, and $\gamma = 3.0$.

5.2 Community Analysis of Exploited Zero-day Vulnerabilities

Three major community archetypes can be identified at a macro level. First, several large and medium-sized communities are structured around generic exploitation primitives, such as input validation failures, injection vectors, or even authentication and access control workarounds. These clusters tend to have a high dispersion between products and suppliers, suggesting that they represent reusable attacker techniques rather than ecosystem-specific weaknesses. Secondly, a set of communities is primarily structured around specific technology ecosystems, notably Microsoft Windows, Apple's platforms, Google Chrome, or Qualcomm firmware. In these cases, the product signal dominates the community structure, with high elevation values on specific OS versions or software branches. Finally, a third class of communities is primarily actor- or technique-focused, where no single product dominates.

- Community 9 (n=29): it aggregates zero-days that are actively exploited and highly severe, dominated by authentication and access control flaws (CWE-287 and CWE-284), exploitable remotely without interaction and with high impacts. Discriminants and lifts converge towards identity abuse, account manipulation and methods of bypassing access mechanisms. Targeted products remain scattered, indicating a community defined by attack techniques rather than a supplier ecosystem.
- Community 6 (n=37): it captures a coherent family of browser-centric exploitation chains dominated by use-after-free and lifecycle errors (CWE-416**,** CWE-825). Although Google Chrome is the primary anchor, the community also includes Linux distributions that propagate the same vulnerable components. The exploitation pattern combines remote attack vectors, low complexity, and frequent user interaction, forming a well-defined "web exploitation" cluster with strong technical homogeneity and high operational impact.
- Community 17 (n=20): it groups exclusively exploited zero-days characterised by high-severity impacts and attack paths requiring no user interaction. The community is structured around flaws affecting execution ordering, resource synchronisation and control-flow enforcement, which are leveraged to reliably escalate privileges in post-compromise scenarios. These vulnerabilities typically require a precise execution context and exploit subtle timing or logic weaknesses rather than broad, opportunistic vectors. Multiple attacker groups recur within the community, indicating sustained and coordinated interest rather than isolated incidents. A notable aspect is the prevalence of low EPSS scores despite confirmed exploitation, suggesting that these technically demanding attack chains tend to be underestimated by probabilistic models. Targeted products remain dispersed, reinforcing the interpretation of a technique-driven community rather than one centred on a specific technology ecosystem.

5.3 UMAP Projection

To compare the community structure obtained from the graph-based modelling with the original contextual representation of the dataset, we project zero-day vulnerabilities using UMAP (Uniform Manifold Approximation and Projection). This visualisation aims to assess whether vulnerabilities sharing similar contextual attributes naturally form clusters and how this organisation relates to the communities identified by the Leiden algorithm. UMAP is a nonlinear dimension reduction method that projects high-dimensional data into a lower-dimensional space while preserving the topological structure of the data.

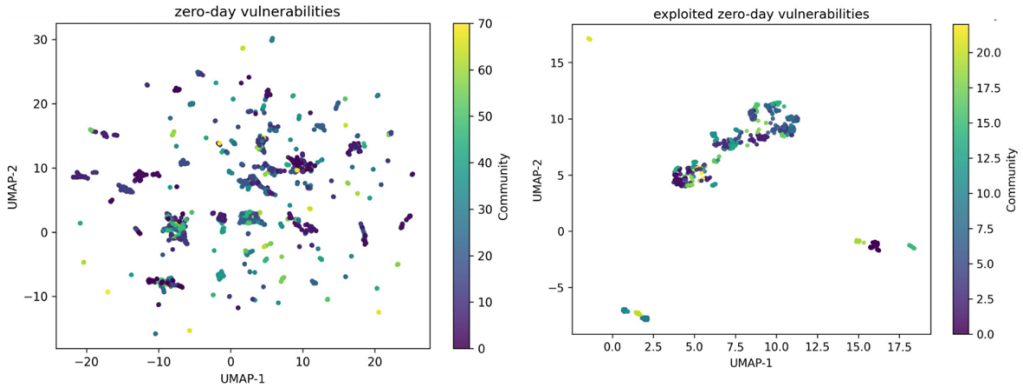

Figure 4: UMAP projection of the raw contextual matrix of exploited zero-day vulnerabilities, coloured by Leiden

These projections (Figure 5) highlight the fact that sharing similar contextual attributes does not necessarily lead to coherent clusters of zero-day vulnerabilities. Local clusters in the raw context space often contain vulnerabilities from multiple communities, illustrating the limitations of similarity based solely on contexts. In contrast, the graph-based community detection approach captured higher-order structural patterns, enabling the identification of meaningful zero-day clusters that are not directly observable in the original context space.

6. Conclusion

This paper introduced a graph-based methodology to analyse and contextualise zero-day vulnerabilities beyond traditional severity-centric approaches. By modelling zero-days as nodes embedded in a heterogeneous graph, we demonstrated that meaningful structural communities emerge without relying on a priori taxonomies or vendor-centric assumptions. The results show that zero-day vulnerabilities do not organise themselves around a single dominant dimension. Instead, communities arise from the interaction between technical weaknesses, exploitation mechanisms and ecosystem contexts, with severity and exploitability scores acting as modulators rather than primary drivers. In particular, high CVSS scores frequently coexist with low or moderate EPSS values, even in communities where exploitation is confirmed, highlighting a systematic mismatch between probabilistic exploitability models and real-world attacker behaviour. The comparison between the full zero-day corpus and the subset of vulnerabilities known to be exploited in the wild further emphasises this gap. While the global corpus is dominated by non-exploited vulnerabilities with generic technical characteristics, the exploited-only graph reveals a fragmented and heterogeneous landscape structured around distinct exploitation logics. Graph-based representations provide a complementary perspective, enabling the identification of latent risk structures, operationally relevant clusters and under-estimated attack chains, particularly in the context of zero-day vulnerabilities. From an operational standpoint, this approach opens new perspectives for vulnerability management, threat-informed prioritisation and risk modelling: By shifting the focus from individual vulnerabilities to communities of related weaknesses, exploitation patterns or technology ecosystems relevant to a given operational context, defenders can better anticipate attacker behaviour and allocate remediation efforts more effectively. Future work will focus on extending this methodology through dynamic and temporal graph representations, not limited to zero-day vulnerabilities, as well as through predictive modelling based on graph learning techniques. Integrating additional threat intelligence sources and evaluating the approach in industrial and cloud environments will further strengthen its applicability to real-world vulnerability management workflows.

Ethics and AI Statement: This research did not require ethical approval. AI-based tools were used solely to facilitate linguistic and editorial improvements; all scientific content, analyses, and conclusions are the sole responsibility of the author.

References

Ablon, L. & Bogart, A. (2017) Zero days, thousands of nights: the life and times of zero-day vulnerabilities and their exploits. Santa Monica, CA: RAND Corporation.

Alharbi, H., et al. (2025) Enhancing cybersecurity through autonomous knowledge graph construction by integrating heterogeneous data sources. PeerJ Computer Science 11:e2768.

Bilge, L. & Dumitras, T. (2012) Before we knew it: an empirical study of zero-day attacks in the real world. ACM conference on Computer and communications security. pp.833-844.

CAPEC MITRE. [online] Available at: https://capec.mitre.org/about/ [Accessed 16 Dec. 2025].
CISA (2020). [online] Cisa.gov. Available at: https://www.cisa.gov/ [Accessed 16 Dec. 2025].
Cover, T. & Hart, P. (1967) Nearest neighbor pattern classification. IEEE Transactions on Information Theory, 13(1), pp.21–27.
CVE (2025). [online] Available at: https://www.cve.org/ [Accessed 16 Dec. 2025].
FIRST (2015). [online] Available at: https://www.first.org/ [Accessed 16 Dec. 2025].
Known Exploited Vulnerabilities Catalog | CISA (2023). [online] Available at: https://www.cisa.gov/known-exploited-vulnerabilities-catalog [Accessed 16 Dec. 2025].
Common Vulnerability Scoring System SIG (2019). [online] Available at: https://www.first.org/cvss/ [Accessed 16 Dec. 2025].
EPSS User Guide (2021a). [online] Available at: https://www.first.org/epss/user-guide [Accessed 16 Dec. 2025].
The EPSS Model (2021b). [online] Available at: https://www.first.org/epss/model [Accessed 16 Dec. 2025].
CPE MITRE (2013). [online] Available at: https://cpe.mitre.org/about/ [Accessed 16 Dec. 2025].
CWE MITRE (2023). [online] Available at: https://cwe.mitre.org/ [Accessed 16 Dec. 2025].
MITRE ATT&CK (2025). [online] Available at: https://attack.mitre.org/ [Accessed 16 Dec. 2025].
NIST NVD (2019). [online] Nist.gov. Available at: https://nvd.nist.gov/ [Accessed 16 Dec. 2025].
Roumani, Y. (2021) Patching zero-day vulnerabilities: an empirical analysis. Journal of Cybersecurity. 2021. 1-13.
Salton, G. & Buckley, C. (1988) Term-weighting approaches in automatic text retrieval. Information Processing & Management, 24(5), pp.513–523.
Shet, A., & Alsmadi, I. (2025) An empirical analysis of zero-day vulnerabilities disclosed by the zero day initiative. arXiv 2512.15803.
Shi, C., et al. (2017) A survey of heterogeneous information network analysis. IEEE Transactions on Knowledge and Data Engineering, pp.17–37.
Traag, V.A., Waltman, L. & van Eck, N.J. (2019) From Louvain to Leiden: guaranteeing well-connected communities. Scientific Reports, 9, Article 5233.
UMAP documentation. [online] Available at: https://umap-learn.readthedocs.io/en/latest/ [Accessed 16 Dec. 2025].
Yin, J., et al. (2024) A compact vulnerability knowledge graph for risk assessment. ACM Trans. Knowl. Discov. Data 18, 8, Article 194, 17 pages.
Zero Day Initiative. [online] Available at: https://www.zerodayinitiative.com/advisories/published/ [Accessed 16 Dec. 2025].