Characterizing License Practices in Maven Central and Their Relationship with CVEs: A CS846 Course Project Proposal

Haonan Zhang
haonan.zhang@uwaterloo.ca
University of Waterloo
Waterloo, Canada

Christina Li
christina.li1@uwaterloo.ca
University of Waterloo
Waterloo, Canada

Paul Wooseok Lee
w69lee@uwaterloo.ca
University of Waterloo
Waterloo, Canada

1 Introduction

Open-source software has been widely used in modern software ecosystems, facilitating the development of various applications, from experimental prototypes to mission-critical organizational platforms [4]. Platforms like Maven Central [6], one of the largest repositories of open-source artifacts, serve as centralized hubs for hosting, managing, and distributing these resources. Most Maven-hosted projects include licenses that provide legal terms and conditions for permissible use, modification, and redistribution, ensuring compliance and clarifying obligations for the users [1]. Apart from licensing, the security of these artifacts is also critical as vulnerabilities in widely used dependencies can pose systemic risks. To facilitate the address of these risks, Common Vulnerabilities and Exposures (CVEs) are frequently reported by users, promoting stakeholders to identify and fix security flaws in a timely manner [8].

Although there have been many studies regarding the vulnerability and license of the artifacts in Maven Central, most of them either focus only on characterizing and addressing the CVEs [2, 12–14] or only on the license adoption and compliance [3, 7, 10, 11]. To the best of our knowledge, there is a lack of studies on the relationships between the licenses adopted by artifacts and their associated CVEs. The variety of licenses not only determines how code can be modified and redistributed but can also influence the community and governance around a project. Some licenses may encourage broader collaboration or more efficient vulnerability reporting processes, while others may present certain barriers to rapid patching and distribution. This raises the hypothesis that license usage patterns could correlate with a project's vulnerability. To fill this gap, we systematically analyze license information and CVE data associated with Maven Central packages, aiming to shed light on whether certain license types are statistically linked to higher or lower incidences of vulnerabilities. Uncovering such patterns can guide developers and stakeholders to make more informed decisions about artifact and license adoption and keep a balance between permissiveness and security.

2 Research Questions

To understand the relationships between the license usages and CVE characteristics, we formulate and address the following research questions:

RQ1: What are the characteristics and trends of license adoption and CVE incidence across Maven Central artifacts? In this research question, we aim to investigate the patterns and trends in how licenses are adopted and how security vulnerabilities (CVEs) manifest within Maven Central artifacts. By analyzing historical and current data, we seek to characterize the prevalence and distribution of different license types as well as the frequency and severity of reported CVEs.

RQ2: Do specific license types correlate with higher or lower vulnerability incidence in Maven Central artifacts? In this research question, we investigate whether choosing a permissive or restrictive license is statistically associated with the incidence of CVEs identified therefore influencing the security of the software.

3 Datasets and Tools

Building upon the Goblin framework, which offers a Neo4j-based Maven Central dependency graph enriched with CVE information, we will extend its ecosystem data by incorporating license metadata for Maven artifacts. Goblin's Weaver tool enables on-demand metric weaving into the existing dependency graph, allowing us to dynamically generate and query additional metrics with vulnerability data. To obtain accurate license details, we will use Libraries.io [5]—queried through a Python script that iterates over relevant Maven artifacts—and merge these results with Goblin's existing Neo4j graph. By combining Goblin's built-in CVE coverage, Neo4j's flexible graph queries and Weaver's on-demand metric computations, we can perform a comprehensive analysis of Maven Central's dependencies, focusing on the interaction between license practices and security vulnerabilities at scale.

4 Schedule and Milestones

- **Week 1:** Extract Maven dependency data with CVE from Goblin and collect and integrate license data from Libraries.io.

- **Week 2-3:** Analyze CVE incidence trends across Maven artifacts and license adoption trends.
- **Week 4-5:** Investigate the statistical relationship between license types and CVEs.
- **Week 6:** Document methodology, results, and findings as a final report.

5 Threats

Rate Limits and Data Incompleteness:
Description: Libraries.io imposes rate limits on API queries, which may cause excessive time in retrieving data, hence leads to partial or failed retrieval of license data for Maven Central artifacts. If these limits are exceeded, the data collection phase may produce incomplete results.
Mitigation: We plan to implement incremental and batched data queries, along with retry strategies to handle temporary rate limit responses. If these measures prove insufficient, we will switch to alternative data sources, such as Sonatype OSS Index [9], or publicly available datasets on licensing information, while acknowledging that they may be less comprehensive or out of date.
Mapping Between Dependencies, Licenses, and Vulnerabilities:
Description: Goblin includes data on Maven Central dependencies and CVEs, whereas the license information is retrieved from external sources. This can introduce mismatches in artifact coordinates or version numbers, leading to incomplete or inaccurate mappings.
Mitigation: We will rely on strict coordinate matching (groupId, artifactId, version) to ensure consistency. In case of partial or ambiguous matches, we will remove such entries from our dataset to reduce noise and maintain high data quality.

6 Conclusion

In this proposal, we outlined a plan to study the interplay between license practices and security vulnerabilities within the Maven Central ecosystem. By leveraging the Goblin framework, which provides dependency and CVE data in a Neo4j graph, and augmenting it with license information from Libraries.io, we aim to discover whether certain license types correlate with higher or lower incidences of reported CVEs. Our schedule includes data collection and integration, trend analysis, and statistical correlation tests, culminating in a final report that summarizes key insights. We anticipate that our work will guide developers and stakeholders in balancing openness with security considerations and contribute new perspectives on how licensing choices influence software ecosystems.

References

[1] Daniel A. Almeida, Gail C. Murphy, Greg Wilson, and Michael Hoye. 2019. Investigating whether and how software developers understand open source software licensing. *Empirical Softw. Engg.* 24, 1 (Feb. 2019), 211–239. https://doi.org/10.1007/s10664-018-9614-9

[2] Johannes Düsing and Ben Hermann. 2022. Analyzing the Direct and Transitive Impact of Vulnerabilities onto Different Artifact Repositories. *Digital Threats* 3, 4, Article 38 (Feb. 2022), 25 pages. https://doi.org/10.1145/3472811

[3] Daniel M. German, Massimiliano Di Penta, and Julius Davies. 2010. Understanding and Auditing the Licensing of Open Source Software Distributions. In *Proceedings of the 2010 IEEE 18th International Conference on Program Comprehension (ICPC '10)*. IEEE Computer Society, USA, 84–93. https://doi.org/10.1109/ICPC.2010.48

[4] Xuetao Li, Yuxia Zhang, Cailean Osborne, Minghui Zhou, Zhi Jin, and Hui Liu. 2025. Systematic Literature Review of Commercial Participation in Open Source Software. *ACM Trans. Softw. Eng. Methodol.* 34, 2, Article 33 (Jan. 2025), 31 pages. https://doi.org/10.1145/3690632

[5] Libraries.io. 2023. *Libraries.io: The Open Source Discovery Service.* https://libraries.io/ [Online; accessed 21-February-2025].

[6] Frederic P. Miller, Agnes F. Vandome, and John McBrewster. 2010. *Apache Maven.* Alpha Press.

[7] Petr Pícha and Souhaila Serbout. 2024. On the Adoption of Open Source Software Licensing - A Pattern Collection. In *Proceedings of the 29th European Conference on Pattern Languages of Programs, People, and Practices (EuroPLoP '24)*. Association for Computing Machinery, New York, NY, USA, Article 19, 7 pages. https://doi.org/10.1145/3698322.3698341

[8] Henrik Plate, Serena Elisa Ponta, and Antonino Sabetta. 2015. Impact assessment for vulnerabilities in open-source software libraries. In *Proceedings of the 2015 IEEE International Conference on Software Maintenance and Evolution (ICSME) (ICSME '15)*. IEEE Computer Society, USA, 411–420. https://doi.org/10.1109/ICSM.2015.7332492

[9] Sonatype. 2025. *Sonatype OSS Index.* https://ossindex.sonatype.org/ [Online; accessed 21-February-2025].

[10] Christopher Vendome, Mario Linares-Vasquez, Gabriele Bavota, Massimiliano Di Penta, Daniel M. German, and Denys Poshyvanyk. 2015. When and why developers adopt and change software licenses. In *Proceedings of the 2015 IEEE International Conference on Software Maintenance and Evolution (ICSME) (ICSME '15)*. IEEE Computer Society, USA, 31–40. https://doi.org/10.1109/ICSM.2015.7332449

[11] Jiaqi Wu, Lingfeng Bao, Xiaohu Yang, Xin Xia, and Xing Hu. 2024. A Large-Scale Empirical Study of Open Source License Usage: Practices and Challenges. In *Proceedings of the 21st International Conference on Mining Software Repositories* (Lisbon, Portugal) *(MSR '24)*. Association for Computing Machinery, New York, NY, USA, 595–606. https://doi.org/10.1145/3643991.3644900

[12] Yulun Wu, Ming Wen, Zeliang Yu, Xiaochen Guo, and Hai Jin. 2024. Effective Vulnerable Function Identification based on CVE Description Empowered by Large Language Models. In *Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering* (Sacramento, CA, USA) *(ASE '24)*. Association for Computing Machinery, New York, NY, USA, 393–405. https://doi.org/10.1145/3691620.3695013

[13] Yulun Wu, Zeliang Yu, Ming Wen, Qiang Li, Deqing Zou, and Hai Jin. 2023. Understanding the Threats of Upstream Vulnerabilities to Downstream Projects in the Maven Ecosystem. In *Proceedings of the 45th International Conference on Software Engineering* (Melbourne, Victoria, Australia) *(ICSE '23)*. IEEE Press, 1046–1058. https://doi.org/10.1109/ICSE48619.2023.00095

[14] Lyuye Zhang, Chengwei Liu, Sen Chen, Zhengzi Xu, Lingling Fan, Lida Zhao, Yiran Zhang, and Yang Liu. 2024. Mitigating Persistence of Open-Source Vulnerabilities in Maven Ecosystem. In *Proceedings of the 38th IEEE/ACM International Conference on Automated Software Engineering* (Echternach, Luxembourg) *(ASE '23)*. IEEE Press, 191–203. https://doi.org/10.1109/ASE56229.2023.00058