ISSUE BRIEF

APRIL 2024

O$$ Security: Does More Money for Open Source Software Mean Better Security? A Proof of Concept

BY JOHN SPEED MEYERS, SARA ANN BRACKETT, AND STEWART SCOTT

EXECUTIVE SUMMARY

The security of open source software has transformed from a niche technology topic to a matter of broad interest in recent years, including for the national security community. Underlying this interest is an emerging consensus that too many users and beneficiaries of open source software are free riders, devoting little money, staff time, and other resources to the health and security of the open source software they depend on, leaving over-burdened and under-compensated open source software maintainers to deal with their code and security fixes, feature improvements, vulnerability remediation, and more on their own. Consequently, this perspective implies that the security and sustainability of open source software suffer from a lack of financial investment. This historical underinvestment has motivated several companies and foundations to recently invest in open source software and associated initiatives.

But is there evidence that more general financial investment ("more money") improves security for open source software projects?

At this project's inception, we could find no existing, large-scale studies on this question,[1] so the project created a novel methodology and dataset to investigate the issue. The project first identified the 1000 most downloaded open source

The **Cyber Statecraft Initiative** works at the nexus of geopolitics and cybersecurity to craft strategies to help shape the conduct of statecraft and to better inform and secure users of technology. This work extends through the competition of state and non-state actors, the security of the internet and computing systems, the safety of operational technology and physical systems, and the communities of cyberspace. The Initiative convenes a diverse network of passionate and knowledgeable contributors, bridging the gap among technical, policy, and user communities.

1 Some funding entities do release limited studies on their efficacy. For examples, see "Tidelift Open Source Maintainer Impact Report," Tidelift, June 2023, https://4008838.fs1.hubspotusercontent-na1.net/hubfs/4008838/Tidelift%202023%20OSS%20Maintainer%20Impact%20Report%20(1).pdf and Michael Scovetta and Michael Winser, "Alpha-Omega 2022 Annual Report," OpenSSF, December 2022, https://openssf.org/wp-content/uploads/2022/12/OpenSSF-Alpha-Omega-Annual-Report-2022.pdf.

software packages in the Python and npm (Javascript) programming language ecosystems, two of the largest and most popular open source package repositories. The project used a tool called "funder-finder" to determine a project's funding sources. Funder-finder sources include GitHub sponsors (both organizational and individual), Tidelift, Open Collective, Google Summer of Code, and NumFOCUS. Additionally, the project used a tool called "Security Scorecard," maintained by the Open Source Security Foundation (OpenSSF), to quantitatively measure the security posture of open source software projects. Finally, the analysis used descriptive statistics and simple statistical procedures to search for evidence of any relationship between funding and security among these popular packages.

Three research findings stand out:

1. The statistical evidence, which is of only moderate strength, supports the view that some general-purpose open-source funding vehicles do correlate moderately with more secure open-source software projects. In short, more money does seem to correlate with better security practices.

2. Additionally, there is some evidence that a greater number of *unique* sources of open source funding for a given project also corresponds to a project with better security practices.

3. The detailed quantitative evidence suggests that more funding positively correlates with better compliance with *several*, thought not all, security practices (as measured by the OpenSSF Security Scorecard tool), instead of any single security practice.

Confidence in these research results must remain only moderate, though, until there is additional research and analysis. These findings rely on cross-sectional data—data at one point in time—and data from only two open source software ecosystems and a small set of funding mechanisms. A later section in the paper describes the analytical limits of this analysis and opportunities for future research.

However, these findings should nonetheless enrich the debate about open source software funding and security. Most importantly, this study presents *prima facie* evidence of a positive effect of general open source software funding on open source software security. This can help funding organizations—companies, non-profits, or governments—make decisions related to funding open source software projects. The findings also suggest that the security effects of funding can be found via relatively straightforward automated analysis and do not strictly require manual data collection from the project maintainers themselves or the invention of new security measurement tools. This can help inform parties that want to evaluate the effects of open source project funding on security to ensure that these dollars are well spent.

Finally, this project—itself more a proof of concept than the final word—highlights several questions for researchers and policymakers interested in open source software funding and security. To mention only a few: How should projects be selected for funding? What is the definition of "security" for an open source project? Can only randomized controlled trials ascertain the true security effects of open source project funding? How well do security practices reduce or prevent negative security outcomes?

In the meantime, interested parties will likely need to adopt the "rough consensus and running code"[2] intellectual style often associated with the open source movement to make sense of the open source project funding and security landscape. That mindset applied to these results leads to a first-cut answer of the main research question of whether more money leads to better open source software security: kind of.

INTRODUCTION: OPEN SOURCE SOFTWARE, MONEY, AND SECURITY

Among those who pay attention to opensource software—code released under licenses allowing anyone to use, inspect, and modify it—there increasingly exists a view that open-source software receives too little investment.[3] Those with this view, moreover, often subscribe to the belief that such under-investment poses substantial risks for much, if not all, of society. One argument considers open-source software as infrastructure, like roads and bridges, or even as critical infrastructure like water reservoirs or hospitals.[4] No matter the exact framing, this reasoning maintains that open source software is ubiquitous throughout the modern digital economy and that risks in open source software—whether arising from unintentional or malicious security concerns, the so-called "health" of the project, or any of other sources—therefore pose dangers to everyone.

2 Andrew L. Russel, "'Rough Consensus and Running Code' and the Internet-OSI Standards War," *IEEE Annals of the History of Computing* (2006), vol. 28, no. 3, 48-61, https://courses.cs.duke.edu/common/compsci092/papers/govern/consensus.pdf.

3 See any of the following: Chris Grams, "Maintainers to industry: We don't have the time nor money to do more," Tidelift, May 11, 2023, https://blog.tidelift.com/maintainers-to-industry-we-dont-have-the-time-nor-money-to-do-more; James Turner, "Open source has a funding problem," January 7, 2021, https://stackoverflow.blog/2021/01/07/open-source-has-a-funding-problem/; Paul Sawers, "Why Sequoia is funding open source developers via a new equity-free fellowship," TechCrunch, February 15, 2024, https://techcrunch.com/2024/02/15/sequoia-open-source-fellowship-developer-funding/; Nadia Eghbal, "Roads and Bridges: The Unseen Labor Behind Our Digital Infrastructure," 2016, The Ford Foundation, https://www.fordfoundation.org/work/learning/research-reports/roads-and-bridges-the-unseen-labor-behind-our-digital-infrastructure/.

4 Eghbal, "Roads and Bridges"; Stewart Scott, Sara Ann Brackett, Trey Herr, Maia Hamin, "Avoiding the success trap: Toward policy for open-source software as infrastructure," The Atlantic Council, February 8, 2023, https://www.atlanticcouncil.org/in-depth-research-reports/report/open-source-software-as-infrastructure/.

This partially explains the Cambrian explosion-like outbreak of open-source software project funding from key players like the Ford Foundation,[5] GitHub Sponsors,[6] the Linux Foundation and associated sub-organizations such as the Open Source Security Foundation (OpenSSF),[7] the National Science Foundation's Pathways to Enable Open-Source Ecosystems (POSE) funding,[8] Open Collective,[9] the Open Technology Fund,[10] the Sovereign Tech Fund,[11] thanks.dev,[12] Tidelift,[13] Spotify,[14] and many others. These funders pursue a variety of goals. Some allow any party with a bank account to simply provide money—no strings attached—to open-source software maintainers.[15] Others are less altruistic and seek social, political, or commercial goals from their funding, such as ensuring that a set of open source software projects critical to their organization's business follow standard engineering and security practices.

The logic of providing funding to open source software maintainers is, at first glance, straightforward. Open source software projects are often maintained by volunteers. While these volunteers are sometimes paid staff from a company dependent on a project, this is usually not the case, and the relative impact of these individuals is well understood. Crucially, this also means that there is often no formal contractual relationship between the parties that depend on the software and the parties that produce and maintain the software.[16] As such, open source software maintainers are often stretched thin.[17] Funding is meant to ease this situation, enabling otherwise busy maintainers to devote more time to their projects.

A related logic is also embedded in this strategy: funding should go to certain "critical" projects, those of particular importance or "central" to a wide number of other entities. Such prioritization, in theory, should ensure that scarce funding dollars are allocated efficiently since there are millions of open-source projects, many of which are no more than personal pet projects.

While these ideas are sensible, there is a tautological danger lurking within. If the "problem" of open source software funding is defined simply as "there is too little funding," then the obvious solution is more funding, more types of it, and more mechanisms for delivering it. Success then becomes defined as many "critical" projects receiving funding, regardless of whether that funding helps improve the projects. While there are certainly moral and ethical arguments to be made in support of such an arrangement, this mindset may ultimately produce disappointing results for advocates interested in changing the underlying projects by funding them. Without a clear sense of purpose, funding could dry up or jump aimlessly from initiative to initiative and project to project. It might also have no discernible effect in the absence of tailored mechanisms to improve security.[18] Ideally, funders of open source software initiatives would state clear goals and use assessment tools[19] similar to those used by international development or medical professionals to evaluate the efficacy of their funding.

To avoid this risk, this research project set out to pilot a relatively formal analysis and measuring of the effect of open source software funding on project security at scale.

So, why security? Security has become a prominent concern among those interested in open source software. Though this movement pre-dates the infamous December 2022 Log4j incident,[20] where a severe vulnerability was found in a ubiquitous open-source logging framework, it certainly gained steam in the aftermath of that major revelation. Key open source software organizations like the Eclipse Foundation and OpenSSF, as well as major companies such as Google, have become evangelists of open source software security. This activity has also spread to the US government (and others internationally), which has led to the US Cybersecurity and Infrastructure Security Administration (CISA)

5 "Critical Digital Infrastructure Research," The Ford Foundation, 2020, https://www.fordfoundation.org/work/learning/learning-reflections/critical-digital-infrastructure-research/.

6 "GitHub Sponsors," https://github.com/sponsors.

7 Open Source Security Foundation, https://openssf.org/.

8 "Pathways to Enable Open-Source Ecosystems (POSE)," US National Science Foundation, 2024, https://new.nsf.gov/funding/opportunities/pathways-enable-open-source-ecosystems-pose.

9 Open Collective, https://opencollective.com/.

10 Open Technology Fund, https://www.opentech.fund/.

11 The Sovereign Tech Fund, https://www.sovereigntechfund.de/.

12 thanks.dev, https://thanks.dev/home.

13 Tidelift, https://tidelift.com/

14 Per Ploug, "Announcing the Spotify FOSS Fund," Spotify, April 22, 2022, https://engineering.atspotify.com/2022/04/announcing-the-spotify-foss-fund/.

15 Most open-source software licenses include a clause distributing the software "as is," protecting contributors from liability for issues arising from the use of the software. Some conceptions of this licensing consider it an exchange—that the price of using code with no financial cost is assuming all liability for any issues within the code. For more, see "History of the OSI," OSI, September 19, 2006, https://opensource.org/history; "Legal Disclaimer and Notices," Github, https://opensource.guide/notices/; and Thomas Depierre, "I Am Not a Supplier," Software Maxims, December 31, 2022, https://www.softwaremaxims.com/blog/not-a-supplier.

16 Depierre, "I Am Not a Supplier."

17 Chris Grams, "Maintainer burnout is real. Almost 60% of maintainers have quit or considered quitting maintaining one of their projects," Tidelift, May 25, 2023, https://blog.tidelift.com/maintainer-burnout-is-real.

18 John Speed Meyers and Jacqueline Kazil, "How to 'harden' open-source software," Binding Hook, November 7, 2023, https://bindinghook.com/articles-binding-edge/how-to-harden-open-source-software/.

19 Alpha-Omega Engagements, https://github.com/ossf/alpha-omega/tree/main/alpha/engagements/.

20 "Review of the December 2023 Log4j Event," CISA Cyber Safety Review Board, July 11, 2022, https://www.cisa.gov/sites/default/files/publications/CSRB-Report-on-Log4-July-11-2022_508.pdf.

developing an Open Source Software Security Roadmap and the establishment of the interagency Open Source Software Security Initiative (OS3I).[21]

This research sought to uncover whether general funding—funding not tied to a specific security goal—measurably improves the security of open source software projects. While the next section will explain the methodological details, it is worth explaining the project's focus on and assumptions around "general" funding. Broadly, the question of whether targeted funding can improve security is both easier to answer and less relevant to the discussion. There is already some evidence that initiatives specifically focused on improving open source software security can succeed. The US Department of Homeland Security's 2006-2009 Open Source Hardening Project is one example,[22] and simply paying for security audits or tooling is a viable approach already piloted by some organizations.[23] It is less clear, however, whether unrestricted or general funds also have security benefits for open source projects. This question is a key part of policy conversations about open source software security—namely *how* resources should be spent on improving security compared to other dependency attributes and *which projects* should receive said support are critical to efficient policymaking.

RESEARCH METHODOLOGY: OR HOW TO EMBRACE "RUNNING CODE"

There is no established body of datasets or techniques for studying the effect of open source software funding on security. One recent study by the company Tidelift that evaluates the impact of its own funding on open source project security is, to our knowledge, the only exception.[24] Beyond the fact that interest in this topic is fairly recent, there are practical reasons for the current research gap. First, the heterogeneous and sprawling nature of open source software means that there is no central authority for funding data, so collection requires either relatively manual processes or new tools. Second, the definition of the "security" of an open source software project—and most software and systems in general—is difficult to pin down, at least for quantitative study. Fourth, even should a researcher create a dataset that tracks funding and "security" for some set of open-source software projects, determining from a methodological standpoint whether that funding *causes* improved security is by no means simple. This research nonetheless attempts to overcome some of these hurdles and answer the question of whether general funding for open source software projects improves their security.

While many might prefer a large-scale randomized controlled trial to answer this research question, such an approach was, for this project, highly impractical. Not only would such a trial require a substantial amount of funding, but it would also be onerous and complicated to administer. Providing financial funding to open source projects not designed to receive funds is also a sensitive matter. In particular, what person or party should receive the funds? This is not always a straightforward question given that many open source projects lack the centralized hierarchy and formal governance that would enable a clear answer. To compensate, this research used regression analysis of cross-sectional data from two open source software ecosystems to measure the statistical relationship between funding and security: the Python Package Index (PyPI, pronounced "pie pea eye") and npm.[25]

PyPI and npm were natural starting points due to their popularity. As of October 2023, PyPI hosts nearly 500,000 open source packages, and npm hosts over one million. These package registries are akin to mobile app stores but house open source packages rather than mobile apps. PyPI has become the go-to source of open source packages for data science, machine learning, artificial intelligence, and other data-related programming activities, making it a central component of modern software. npm, a package registry for Javascript, is the leading package manager for software developers building web applications, both the "front-end" parts visible to a user through a browser and the "back-end" code running on servers. Additionally, these ecosystems have already been the focus of other software security research, which suggested the feasibility of this study.[26] Future analysis can extend this project's analysis to other ecosystems.

The next step of the project involved creating a clear definition of "funding." There are admittedly many types of funding for open source software projects, serving a wide variety of purposes and methods ranging from straightforward transfers of cash to portioning out developer time from large IT firms. Funding, for the purposes of this project, is defined as whether there exists evidence that a particular project has any of the following funding sources:

21 "CISA Open Source Software Roadmap," CISA, September 2023, https://www.cisa.gov/sites/default/files/2023-09/CISA-Open-Source-Software-Security-Roadmap-508c.pdf; "Fact Sheet: Biden-Harris Administration Releases End of Year Report on Open-Source Software Security Initiative," The White House, January 30, 2024, https://www.whitehouse.gov/oncd/briefing-room/2024/01/30/fact-sheet-biden-harris-administration-releases-end-of-year-report-on-open-source-software-security-initiative/.

22 Meyers and Kazil, "How to 'harden' open-source software."

23 Open Source Technology Improvement Fund, https://ostif.org/; Alpha-Omega, https://alpha-omega.dev/; Chris Aniszczyk, "Open sourcing the Kubernetes security audit," Cloud Native Computing Foundation, August 6, 2019, https://www.cncf.io/blog/2019/08/06/open-sourcing-the-kubernetes-security-audit/.

24 Lauren Hanford, "New data showing the impact of paying maintainers to improve open source security," Tidelift, July 20, 2023, https://blog.tidelift.com/new-data-showing-the-impact-of-paying-maintainers-to-improve-open-source-security.

25 Python Package Index, https://pypi.org/; npm, https://www.npmjs.com/.

26 Nusrat Zahan et al., "What are weak links in the npm supply chain?," *ICSE-SEIP '22: Proceedings of the 44th International Conference on Software Engineering: Software Engineering in Practice* (2022): 331–340, https://dl.acm.org/doi/abs/10.1145/3510457.3513044; Duc-Ly Vu, Zachary Newman, and John Speed Meyers, "Bad Snakes: Understanding and Improving Python Package Index Malware Scanning," ICSE '23: Proceedings of the 45th International Conference on Software Engineering (2023): 499–511, https://dl.acm.org/doi/abs/10.1109/ICSE48619.2023.00052.

- Official funding through GitHub Sponsors,[27] an official funding program created by GitHub, for the parent GitHub organization of a project
- Individual funding from GitHub Sponsors for any of the top three contributors to a project
- Funding from Tidelift,[28] a company that matches funding from companies with open-source software maintainers
- Funding from Open Collective,[29] a tool for grassroots fundraising sometimes used by open-source projects as well as by large, centralized funding entities
- Funding from NumFOCUS,[30] a non-profit that, among other things, provides fiscal sponsorship of open source projects related to research, data, and scientific computing
- Funding from Google Summer of Code, a program that provides funding for new contributors to work on open source projects in an internship-like fashion

This study assumes that these are general sources of funding, though caveats and exceptions to this assumption are discussed in the limitations section. These funding parameters were assessed via the open-source tool funder-finder, which uses simple heuristics to determine whether a given open source project (specifically, a GitHub URL) has evidence of these types of funding.[31] Other approaches are possible, though these have their drawbacks. For instance, surveying the developers associated with a set of open source projects is feasible but arduous. Additionally, it would also be possible to partner with a group of funding entities and standardize their data, although this would require significant formal cooperation.

To measure the "security" of an open source project, this research used the OpenSSF Security Scorecard tool,[32] which provides a score from zero to ten to grade the maturity and trustworthiness of a project's security development practices or its security posture. The tool rely on a series of subchecks and heuristics to determine whether a project follows a set of well-known security practices. The benefit of using Scorecard is that the assessment is automated, comparable, and focused on practices rather than outcomes, which are challenging to measure and prone to significant biases. There are other potential approaches for measuring the security posture of an open source project, such as the mean time to remediation for disclosed vulnerabilities, but such datasets tend to be expensive to collect and are often incomplete.

The actual analysis involved measuring the funding and security scores for the 1000 most popular projects in both Python and npm and then comparing their scores against those of projects without funding. The analysis also compared the security scores of projects by funding type. Focusing on only the top downloaded projects increases the likelihood that the funded and unfunded are comparable in terms of organization and scope (with some exceptions). This approach should minimize the risk of comparing major, relatively well-organized projects (that receive funding) to projects that are in the "long tail" of open-source development (and are unfunded), which are in reality no more than minor personal projects.

For both PyPI and npm, the analysis followed these steps:

1. Create a list of the 100 most popular open-source packages. For Python, the most popular packages were defined as those with the most downloads.[33] For npm, the most popular packages were defined according to npm rank, which provided a most-depended-upon list.[34]
2. Identify the source code URL for each project. For the Python packages, the list of URLs was created via the open-source tool deps2repos.[35] For npm, the rankings list also provided URLs.
3. Run funder-finder on all GitHub URLs. Funder-finder can only report results for GitHub URLs. Fortunately for this analysis, most of the referenced projects are hosted on GitHub.
4. Run Scorecard on all GitHub URLs.
5. Create descriptive statistics for funder-finder and Scorecard results.

27 GitHub Sponsors, https://github.com/sponsors.
28 Tidelift, https://tidelift.com/.
29 Open Collective, https://opencollective.com/
30 NumFOCUS, https://numfocus.org/.
31 Funder-finder, Georgetown Center for Security and Emerging Technology (CSET), https://github.com/georgetown-cset/funder-finder.
32 Scorecard, OpenSSF, https://github.com/ossf/scorecard.
33 Top PyPI Packages, https://hugovk.github.io/top-pypi-packages/top-pypi-packages-30-days.json
34 Andrei Kashcha, Top 1000 most depended-upon packages, https://gist.github.com/anvaka/8e8fa57c7ee1350e3491#file-01-most-dependent-upon-md
35 Deps2repos, Open Source Software Neighborhood Watch, https://github.com/Open-Source-Software-Neighborhood-Watch/deps2repos.

RESULTS: DOES MORE MONEY MEAN BETTER OSS SECURITY?

The main results from this analysis are four-fold:

- There are a variety of funding types in both the PyPI and npm ecosystems.
- Some funding types do appear to correlate to substantially higher security posture scores. In particular, GitHub organization sponsorship, Open Collective funding, and, to a lesser extent, Tidelift funding appear to correlate strongly with security benefits. GitHub individual funding does not appear to influence a project's security posture.
- There is moderate evidence that combined funding (i.e. having more unique funding sources) is also correlated with better security posture.
- The detailed quantitative evidence suggests that funding positively correlates with an array of security practices (as measured by the OpenSSF Security Scorecard), meaning that funding does not correlate with a better score in any single security practice alone .

Table 1 provides summary data about the prevalence of funding by ecosystem and funding type.

Most strikingly, the npm ecosystem appears to have relatively more funding than PyPI. For most categories of funding, the number of npm projects with funding in that category is double or triple the number of Python projects with that category of funding. It is also notable that some types of funding such as GitHub Sponsors (especially individual sponsors) are quite common while others, such as NumFOCUS and Google Summer of Code, are rare. This is not to imply that they are insignificant investments into the OSS ecosystem—indeed they might prioritize quality support to a small number of projects with critical or niche uses instead of widespread funding—but rather that for this analysis, the latter two simply provide too small a sample for showing statistical significance. However, several types of funding have sufficient data to enable robust statistical analysis.

Statistical analysis of the Python ecosystem reveals that all forms of funding correlate with an improvement in the security posture of projects (see Table 2).

Table 1. Funding Prevalence by Ecosystem and Funding Type

Funding Type	PyPI	npm
Github Sponsors - individual	243	496
Github Sponsors - organizational	67	53
Tidelift	50	185
Open Collective	36	122
NumFOCUS	11	0
Google Summer of Code	11	21

Table 2. Effect of Funding by Funding Type for the Python Ecosystem

Funding Type	Average Scorecard Score	Statistically Significant Difference from No Funding?
None	5.17	N/A
Any	5.58	YES ($p < .01$)
GitHub Organizational	6.01	YES ($p < .01$)
GitHub Individual	5.52	YES ($p < .01$)
Tidelift	6.55	YES ($p < .01$)
Open Collective	6.36	YES ($p < .01$)

For the Python ecosystem, some forms of funding, especially Tidelift, Open Collective, and GitHub Organizational Sponsorship correlate with a significantly better security score, approximately one point or more higher on average.

Table 3. Effect of Funding by Funding Type for the npm Ecosystem

Funding Type	Average Scorecard Score	Statistically Significant Difference from No Funding?
None	4.3	N/A
Any	4.26	NO
GitHub Organizational	5.68	YES ($p < .01$)
GitHub Individual	4.31	NO
Tidelift	4.34	NO
Open Collective	5.42	YES ($p < .01$)

Table 3 reveals that the funding effects in the npm ecosystem vary. Some types of funding appear to have no effect. But two forms of funding, GitHub Sponsors organizational funding and Open Collective, correlate with a better overall security posture. Additional analysis also suggests that an increase in the number of unique funders also leads to an improvement in the average scorecard for projects in both ecosystems (see Table 4).

Table 4. Average Scorecard Score by Number of Unique Funders for both the Python and npm Ecosystems

Number of Unique Funders	Python	npm
0	5.17	4.21
1	5.25 (n=214)	4.25 (n=585)
2	6.41 (n=60)	4.30 (n=108)
3	6.26 (n=14)	5.35 (n=164)
4	7.42 (n=5)	5.10 (n=11)
5	7.00 (n=2)	N/A (n=0)

Python projects with two or more funding sources have noticeably higher average overall security posture scores. npm projects with three or more funding sources also appear to have higher average security posture scores. This analysis suggests that there are potentially distinct benefits from having more unique sources of funding on top of a project having a source of funding in general.

SUBCHECKS

OpenSSF Security Scorecard scores are a composite of several different subscores, each produced from assessing a project's adherence to some security practice given a specific weight for the final calculation. Looking at which of these subchecks drove variations in overall scores highlights the details of the funding-scorecard relationship. Table 5 summarizes these for Python and Table 6 for npm, providing the difference for each subcheck between the mean scores of each funder and the mean scores of unfunded projects.

The following practices were significantly more common in Python projects with any funder:

- Continuous Integration (CI) tests
- Core Infrastructure Initiative (CII) best practices
- A diversity of recently active company-affiliated contributors
- Avoidance of dangerous workflows
- Fuzzing tools
- Active maintainers
- Official package building practices
- Cryptographical signatures on releases
- Read-only permissions on GitHub workflow tokens
- Remediation of known vulnerabilities.

Some practices were significantly more common among specific funders. Projects with Open Collective funding were more likely to review code before merging it, and projects with either Open Collective or Tidelift support were significantly more likely to use dependency update tools and have security policies. Tidelift-supported projects were also much more likely to use static code analysis tools.

Table 5. Scorecard Subchecks Across Funders in PyPI - Differences from No-funder Average (* = p<.05)

Check Type	Any Funder	Tidelift	Open Collective	GitHub Individual	GitHub Organizational
Binary Artifacts	-0.13	-0.12	+0.18	-0.12	-0.75*
Branch Protection	-0.34	-0.41	+0.49	-0.42	+0.17
CI Tests	**+1.14***	**+2.63***	**+3.16***	**+0.91***	**+2.38***
CII Best Practices	**+0.19***	**+0.89***	**+0.53***	**+0.13***	**+0.23***
Code Review	-0.03	-0.68	**+2.52***	**-0.59***	**+1.43***
Contributors	**+0.82***	**+1.24***	**+1.26***	**+0.74***	**+1.30***
Dangerous Workflow	**+1.33***	**+2.66***	**+1.09**	**+1.22***	**+2.22***
Dependency Update Tool	+0.66	**+4.03***	**+2.64***	+0.64	**+1.45***
Fuzzing	**+1.23***	**+1.20***	**+2.80***	**+1.13***	**+1.93***
License	+0.03	+0.20	-0.16	+0.08	-0.13
Maintained	**+1.49***	**+3.83***	**+4.14***	**+1.33***	**+2.54***
Packaging	**+0.69***	**+1.60***	**+1.60***	**+0.87***	+0.13
Pinned Dependencies	**-0.34***	**-0.90***	**-1.58***	-0.16	-0.33
SAST	+0.23	**+2.00***	+0.68	+0.21	+0.30
Security Policy	+0.49	**+3.79***	**+1.99***	+0.45	+0.99
Signed Releases	**+0.35***	+0.18	**+0.62***	**+0.45***	**+1.03***
Token Permissions	**+0.96***	**+3.66***	**+0.94***	**+0.99***	**+1.07***
Vulnerabilities	**+0.44***	+0.54	**+0.31***	**+0.47***	+0.49

For npm projects, the following were more common among projects with any source of funding:

- A diversity of recently active company-affiliated contributors
- An absence of dangerous workflows
- Security policies
- Remediation of known vulnerabilities

Oddly, dependency update tools and branch protection were significantly *less* common among funded npm projects. Some practices were significantly more common only among specific funders. Open Collective projects saw more CII best practices, dependency update tools, code review, fuzzing, maintenance, packaging, pinned dependencies, and read-only token permissions. Meanwhile, Tidelift-supported projects saw significantly better vulnerability remediation, presence of security policies, and organizationally-backed contributors. GitHub organizational sponsors were more common in projects with branch protection, CI tests, organizationally-backed contributors, fuzzing, maintenance, packaging, read-only token permissions, and security policies.

Table 6. Specific Scorecard Components Across Funders in npm - Differences from No-funder Average (* - p<.05)

Check Type	Any Funder	Tidelift	Open Collective	GitHub Individual	GitHub Organizational
Binary Artifacts	+0.09	+0.09	+0.04	+0.09	+0.06
Branch Protection	**-0.80***	**-1.32***	+0.62	**-0.96***	**+1.13***
CI Tests	-0.45	**-1.38***	**+2.70***	**-0.73***	**+3.00***
CII Best Practices	+0.00	-0.04	**+0.29***	-0.02	+0.13
Code Review	-0.10	-0.41	**+1.39***	-0.21	+0.77
Contributors	**+0.52***	**+0.86***	+0.83	**+0.49***	**+0.89***
Dangerous Workflow	**+1.48***	**+3.19***	**+3.11***	**+1.33***	**+2.81***
Dependency Update Tool	**-2.19***	**-3.74***	**+2.13***	**-2.69***	+0.02
Fuzzing	+0.05	+0.05	**+0.58***	-0.04	**+0.71***
License	+0.17	+0.25	+0.15	+0.18	+0.30
Maintained	**-0.24***	**-0.98***	**+4.01***	-0.51	**+2.26***
Packaging	+0.12	-0.13	**+0.85***	+0.10	**+0.98***
Pinned Dependencies	**-0.10***	**-0.33***	**+1.06***	**-0.25***	+0.26
SAST	**-0.45***	**-0.65***	+0.64	**-0.54***	+0.18
Security Policy	**+2.20***	**+6.50***	**+1.97***	**+2.03***	**+5.53***
Signed Releases	**+0.01***	**-0.04***	+0.06	+0.00	**+0.36***
Token Permissions	+0.42	+0.10	**+1.86***	+0.30	**+1.07***
Vulnerabilities	**+0.55***	**+2.35***	**-1.84***	**+0.79***	-0.25

In short, these findings indicate with moderate confidence that there is a meaningful connection between more open-source project funding and improved security posture. Some practices are strongly associated with funding, and more funding generally correlates with more dramatically differentiated security practices. Not all funders had scores that indicated significantly differentiated security practices in all ecosystems, but all did have a significant number of subchecks with dramatic score improvements correlating to funding.

If these results indicate that funding leads to better security practices, the causal explanation is relatively simple and intuitive. More money for maintainers and even developers means more flexibility in dedicating time to project management, which can include developing security policies, remediating vulnerabilities, using better permission tokens, and including CI testing, fuzzing, signatures, packaging, update tools, and more—all of which increase Scorecard scores. Additionally, funding may help purchase either tooling or project services that would similarly contribute to security posture.

LIMITATIONS AND DISCUSSION

There are several limitations to this analysis that are important to acknowledge. First and most significantly, escaping the causality-correlation question is particularly challenging in this space. It seems reasonable that projects with good general practices, including security practices, are more likely to attract funding—or at least as reasonable as the notion that funding helps projects improve their security practices. This logic is particularly salient for some of the subchecks. For example, on the one hand, general funding might allow a project's maintainers to spend more time working on it and enable them to bring on additional maintainer

support, increasing the chances that the project receives a high maintainer subcheck score. On the other, a project with a vibrant maintainer cohort seems reasonably more likely to receive funding by virtue of having administrators in the position to advocate for their project as well as to seek out and receive funds. While strictly disaggregating causation from correlation in this project is out of scope, the next section discusses future research avenues, which include methods of tackling this causation-correlation challenge.

Still, some evidence suggests that funding predates rather than results from better security posture. Tidelift's 2023 Open Source Development Impact Report tracks the Scorecard results of a cohort of Tidelift-supported projects over twenty months, which received direct incentives to improve some Scorecard subcheck scores. The study also compares the cohort's results to the average score of all open-source packages in a vaguely specified peer group that did not receive the incentives treatment. The result was that, over the reporting period, the Scorecard score of the cohort of Tidelift projects steadily increased while the scores of the other open-source projects remained static or even declined.[36] Moreover, the study, which involved direct incentives to improve Scorecard results, also asked maintainers their thoughts on the arrangement and determined that maintainers found the incentives either neutral or better compared to the added compliance burden, and that 55 percent of the cohort was either neutral on continuing the Scorecard work or unlikely to continue without the provided incentives. This may suggest that direct incentivization can drive security posture improvements, although the size of the examined cohort is small (twenty six) and the study's methodology is not fully clear. Even if funding results from, rather than leads to, improved security posture, that would at least create an incentive for maintainers to improve security practices—although this does little to address obstacles to resourcing those changes in the first place.

Relatedly, this study's assumption that all the examined funders are general funders is an oversimplification. Open Collective, for example, serves as a funding conduit, enabling both funders and fund recipients. Some of the funders it enables include large firms such as Salesforce, Morgan Stanley, or Google, as well as other entities outside the private sector, including, to some degree, NUMFocus. Tidelift, similarly, works with maintainers to explicitly improve security practices and requires supported projects to take some measures that might boost their scorecards.[37] One example is the robust association between Tidelift funding and the presence of a project security policy, particularly in the npm ecosystem. Tidelift terms require such a policy, and so long as the file is named SECURITY.md, it will satisfy the Scorecard check. In this way, Tidelift funding is more specific than general funding, but the other requirements it makes of maintainers appear minimal enough to still constitute as general funding for the purposes of this proof-of-concept study. Moreover, the question of whether funding *directly incentivizes* a certain security practice or simply *enables* project maintainers to establish a practice they intended or wished to adopt given enough time and resourcing is out of the scope of this study. More broadly, capturing the full extent of funding obligations is difficult at scale as these criteria might vary significantly from project to project, but this is an issue worthy of future research. Some projects captured by this study are likely both funded and supported by software foundations, which may impose governance obligations on projects or provide additional resourcing or requirements for security—in theory, it would be possible that only foundation support causes improved Scorecard results, but funding is a moderately strong predictor of foundation support among very popular projects.

Finally, the OpenSSF Scorecard tool itself has several quirks relevant to this analysis.[38] First, Scorecard does not necessarily capture all the security improvements that might occur during a project. For example, a security audit, paid for voluntarily by maintainers who received funding, might look for and remediate new vulnerabilities while likely not directly improving any Scorecard metric. Some practices may also meet OpenSSF criteria in principle but be missed by the automated scorecard check. For example, a project might include a strong security policy, but with a different filename than what the scorecard search is looking for, and thus not pass the subcheck. The conversion of subchecks to an overall score also likely dampens some of the stronger correlations in the overall analysis as some scores are weighted less than others.

IMPLICATIONS FOR FUTURE RESEARCH

This pilot study suggests a variety of future research efforts that could enrich the current state of knowledge related to open source software funding and security.

First, expanding the set of analyzed open source software ecosystems would be a clear improvement. This study covered only the PyPI and npm ecosystems. There are, of course, many others. Examining other popular ecosystems, like Maven Central (Java) or RubyGems (Ruby), is one potentially useful next step. Additionally, this study focused only on the top 1000 most popular packages in each ecosystem. Examining fewer of the most popular packages or many more packages could also yield analytical dividends, as could expanding the criteria of what constitutes "important packages" beyond top downloads.

36 "Tidelift Open Source Maintainer Impact Report," Tidelift, June 2023, https://4008838.fs1.hubspotusercontent-na1.net/hubfs/4008838/Tidelift%202023%20OSS%20Maintainer%20Impact%20Report%20(1).pdf.

37 Caitlin Bixby, "Lifter tasks overview," Tidelift, November 2023, https://support.tidelift.com/hc/en-us/articles/4406288074260-Lifter-tasks-overview#h_01HFPT03434FVANGJPS3SMFRTV.

38 Nustra Zahan, et al., "OpenSSF Scorecard: On the Path Toward Ecosystem-wide Automated Security Metrics," arXiv, June 15, 2023, https://arxiv.org/pdf/2208.03412.pdf.

Second, there are many more types of funding for open source software that could be studied beyond the handful analyzed in this report. One particularly fruitful approach could be partnering with one or more organizations that provide open source software funding and using their presumably more detailed and accurate funding datasets. Building on "funder-finder" is another option, as is examining projects supported by the allocation of full-time developer hours from industry or the difference in projects supported by foundations or stewardship models versus other funding structures.

Third, the definition of "security" employed in this study is admittedly narrow and is tied to the OpenSSF Scorecard tool. There are two broad options for expanding and improving on this definition. One is to undertake research that validates the usefulness of the Scorecard tool by examining the relationship between the checks in Scorecard and actual security outcomes. This would be an ambitious but valuable research avenue, directly tying security practice and outcome. Another is to simply leave OpenSSF scorecards behind and build new "security" datasets (without simply reverting to theoretically flawed counts of known vulnerabilities). One possible angle is to focus on the remediation time for known vulnerabilities, though collecting such a comprehensive dataset for open source software projects would be a substantial undertaking on its own.

Fourth, several methodological alternatives could potentially provide a more reliable estimate of the actual causal effect of funding on open-source project security. One relatively straightforward option is research that measures both funding and Scorecard score over time to better approximate causation. Scorecard time-series data is already available for most projects, although gathering data on funding over time is likely an intensive process. A more ambitious approach is to create randomized controlled trials in which funding is actually allocated to projects at random. While this methodology would be the most desirable, there would be a number of operational and practical challenges, although a serious funder may be willing to consider such an expensive evaluation option. Additionally, interviewing maintainers of both funded and unfunded projects could shed light on how receipt of funding changes project practices and outcomes, both generally and those related to security.

Fifth, *why* exactly general open-source software funding correlates with improved software security posture remains an open question. Through what mechanisms does this funding operate and why are they effective? Understanding these mechanisms could potentially allow the design of more effective funding programs.

IMPLICATIONS FOR PUBLIC POLICY

Many readers will likely finish this study with more questions than answers. These questions, however, are key steps toward a more rigorous understanding of how policy can impact security outcomes. First, this study provides *prima facie* evidence that some types of general-purpose open source software funding correlate to better open source project security posture. Even funding that is not specifically aimed at security appears to correlate to better security posture, at least marginally. It seems plausible that open source software funding specifically focused on security is even more likely to have tangible security benefits.[39]

Second, this study also suggests that it is indeed feasible to conduct quantitative evaluations of the effect of open source software funding on open source software security. This could move funders to adopt a new definition of "success" beyond simply considering the disbursement of funds to projects they deem important sufficient in and of itself. Instead, funders can realistically use approaches like this one to measure security improvements from their funding.

Perhaps the most enduring contribution of this study will be to inspire a set of questions that must be implicitly answered by any organization that wants to create large-scale funding of open source software projects that leads to positive security benefits. These questions are inspired by the methodological challenges faced during this study.

First, how should government agencies and other funders select projects? This study focused on only the most popular projects in two widely-used open source ecosystems, but these counts of download or dependency do little to account for the context of a projects use, which significantly impacts the risks that might come from its compromise. They also might not reflect the true spread of a project, which industry reliance on internal repositories might skew. How could a public program make principled decisions about project selection? One option is to solicit funding requests, though this option has the disadvantage that poorly-resourced projects might lack the means to apply. Another option is using data from government agencies about what open source software is most "critical" to their operations, although it is unclear how onerous it would be to create such a dataset given the strenuous task of identifying and updating dependencies and their contexts frequently and at massive scale.

Second, what is the exact definition of security that such a program seeks? Is automated security posture analysis, such as OpenSSF Security Scorecard, adequate? Or is some other type of measurement required? These modest results are

39 Meyers and Kazil, "How to 'harden' open-source software;" Hanford, "New data showing the impact of paying maintainers to improve open source security."

a reason that funders will likely not be able—at least in the short term—to adopt the high-confidence decision-making style sometimes associated with more mature public policy areas. Instead, those interested in open source software funding for security will need to adopt an intellectual style often associated with the open source movement: "rough consensus and running code."[40] Only in this way will they currently be able to make sense of the open source project funding and security landscape. Returning to the question that originally animated this study: does more money for open source software projects correlate to better security posture? Our answer: in many cases, yes, although with modest confidence. What is very clear, however, is the necessity for researchers and policymakers to look more closely at mechanisms designed to incentivize good cybersecurity practices and understand how they do (and do not) drive behavior and outcomes.

ACKNOWLEDGMENTS

We thank Jennifer Melot for being the brains behind "funder-finder," the open source software tool for measuring open source software funding that made this analysis possible. We're also grateful to Trey Herr for nurturing this research agenda. Additionally, we thank Abhishek Arya, Aeva Black, Derek Zimmer and Amir Montazery, and Trey Herr for their valuable feedback on earlier versions of this paper.

ABOUT THE AUTHORS

John Speed Meyers is a nonresident senior fellow with the Atlantic Council's Cyber Statecraft Initiative and the head of Chainguard Labs at Chainguard. He oversees research on open source software security, software supply chain security, and container secu-rity. He previously worked at IQT Labs, the RAND Corporation, and the Center for Strategic and Budgetary Assessments. He holds a PhD in policy analysis from the Pardee RAND Grad-uate School, a master's in public affairs (MPA) from Prince-ton's School of Public and International Affairs, and a BA in international relations from Tufts University.

Sara Ann Brackett is a research associate at the Atlantic Council's Cyber Statecraft Initiative. She focuses her work on open-source software security (OSS), software bills of materials (SBOMs), software liability, and software supply-chain risk management within the Initiative's Systems Security portfolio. Brackett is currently an undergraduate at Duke University, where she majors in Computer Science and Public Policy and is currently writing a thesis on the effects of market concentration on cybersecurity. She participates in the Duke Tech Policy Lab's Platform Accountability Project and works with the Duke Cybersecurity Leadership Program as part of Professor David Hoffman's research team.

Stewart Scott is an associate director with the Atlantic Council's Cyber Statecraft Initiative. He works on the Initiative's systems security portfolio, which focuses on software supply chain risk management and open source software security policy.

40 Russel, "Rough Consensus and Running Code' and the Internet-OSI Standards War."

Atlantic Council

Board of Directors

CHAIRMAN
*John F.W. Rogers

EXECUTIVE CHAIRMAN EMERITUS
*James L. Jones

PRESIDENT AND CEO
*Frederick Kempe

EXECUTIVE VICE CHAIRS
*Adrienne Arsht
*Stephen J. Hadley

VICE CHAIRS
*Robert J. Abernethy
*Alexander V. Mirtchev

TREASURER
*George Lund

DIRECTORS
Stephen Achilles
Elliot Ackerman
*Gina F. Adams
Timothy D. Adams
*Michael Andersson
Alain Bejjani
Colleen Bell
Sarah E. Beshar
Stephen Biegun
Linden P. Blue
Brad Bondi
John Bonsell
Philip M. Breedlove
David L. Caplan
Samantha A. Carl-Yoder
*Teresa Carlson
*James E. Cartwright
John E. Chapoton
Ahmed Charai
Melanie Chen
Michael Chertoff
*George Chopivsky
Wesley K. Clark
*Helima Croft
Ankit N. Desai
Dario Deste
Lawrence Di Rita
*Paula J. Dobriansky
Joseph F. Dunford, Jr.
Richard Edelman
Stuart E. Eizenstat
Mark T. Esper
Christopher W.K. Fetzer
*Michael Fisch
Alan H. Fleischmann
Jendayi E. Frazer
*Meg Gentle
Thomas H. Glocer
John B. Goodman
Sherri W. Goodman
Marcel Grisnigt
Jarosław Grzesiak
Murathan Günal
Michael V. Hayden
Tim Holt
*Karl V. Hopkins
Kay Bailey Hutchison
Ian Ihnatowycz
Mark Isakowitz
Wolfgang F. Ischinger
Deborah Lee James
*Joia M. Johnson
*Safi Kalo
Andre Kelleners
Brian L. Kelly
John E. Klein
*C. Jeffrey Knittel
Joseph Konzelmann
Keith J. Krach
Franklin D. Kramer
Laura Lane
Almar Latour
Yann Le Pallec
Jan M. Lodal
Douglas Lute
Jane Holl Lute
William J. Lynn
Mark Machin
Marco Margheri
Michael Margolis
Chris Marlin
William Marron
Gerardo Mato
Erin McGrain
John M. McHugh
*Judith A. Miller
Dariusz Mioduski
*Richard Morningstar
Georgette Mosbacher
Majida Mourad
Virginia A. Mulberger
Mary Claire Murphy
Julia Nesheiwat
Edward J. Newberry
Franco Nuschese
Joseph S. Nye
*Ahmet M. Ören
Ana I. Palacio
*Kostas Pantazopoulos
Alan Pellegrini
David H. Petraeus
*Lisa Pollina
Daniel B. Poneman
Robert Portman
*Dina H. Powell McCormick
Michael Punke
Ashraf Qazi
Thomas J. Ridge
Gary Rieschel
Charles O. Rossotti
Harry Sachinis
C. Michael Scaparrotti
Ivan A. Schlager
Rajiv Shah
Wendy R. Sherman
Gregg Sherrill
Jeff Shockey
Ali Jehangir Siddiqui
Kris Singh
Varun Sivaram
Walter Slocombe
Christopher Smith
Clifford M. Sobel
Michael S. Steele
Richard J.A. Steele
Mary Streett
Nader Tavakoli
*Gil Tenzer
*Frances F. Townsend
Clyde C. Tuggle
Francesco G. Valente
Melanne Verveer
Tyson Voelkel
Michael F. Walsh
Ronald Weiser
*Al Williams
Ben Wilson
Maciej Witucki
Neal S. Wolin
Tod D. Wolters
*Jenny Wood
Guang Yang
Mary C. Yates
Dov S. Zakheim

HONORARY DIRECTORS
James A. Baker, III
Robert M. Gates
James N. Mattis
Michael G. Mullen
Leon E. Panetta
William J. Perry
Condoleezza Rice
Horst Teltschik
William H. Webster

**Executive Committee Members*

List as of February 6, 2024