How maintainer Jordan Harband saved the popular minimist JavaScript package from deletion
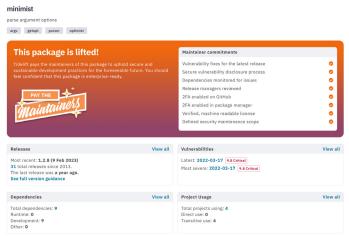
A while back, npm made the decision to start requiring maintainers of JavaScript packages to use two-factor authentication. While many organizations applauded the decision because it would increase package security, some JavaScript maintainers weren't so enthusiastic, because requiring 2FA was just one additional burden on them that many of them weren't being compensated to take on.

For one particular maintainer of several high profile packages, including one called minimist, who went by the name "substack," this new requirement was a problem.

One of Tidelift's maintainer partners, a super maintainer named Jordan Harband, worked regularly with substack, and even helped maintain several of his other projects. One day Jordan was having a conversation with him online, and substack indicated that he was planning on deleting his GitHub account entirely, in part because of the new 2FA requirement. Unusual perhaps, but because of his lifestyle, substack did not own a mobile phone, so 2FA would be extra difficult for him to comply with.

It was clear from Jordan's conversation that the 2FA requirement was [one final unfunded mandate](#) more than substack was willing to take on to support these projects he developed and maintained on a volunteer basis (sadly, maintainers quitting or threatening to quit maintaining their projects because of increasing expectations and inadequate compensation for their work is more common than one might imagine, with [58% of maintainers reporting that they have quit or considered quitting](#) their maintenance work).

Jordan had a moment of panic when he found out that substack planned to delete his entire GitHub account, because many projects Jordan maintains and relies upon were still under substack's username, and a lot of the packages Jordan maintains depend on his packages as well.

Here is the minmist package page inside of the Tidelift application, showing the secure development practices the project is following, and also including additional useful information about releases, vulnerabilities, dependencies, and where it is being used within an example organization.

Substack deleting his GitHub account would instantaneously delete all of the associated GitHub repositories, which would not only impact many of Jordan's own packages that relied on them, but would impact any organization that relied on these packages—which were in broad use. The removal of these packages would have potentially dire consequences.

Jordan offered to take over maintenance on all of the packages, and was able to get substack to transfer a few of them. But when it became clear that there was not an easy "one click" way to transfer ownership of the rest of the packages, substack lost interest and went ahead and deleted his account.

Jordan recalled watching as the packages started to disappear, one by one, until they were all gone. After trying a few other things to see if he could recover or manually re-create them, Jordan reached out to GitHub to see if there was any way to undelete the packages and transfer ownership of them, and called in Tidelift to help make the case for him to take over ownership of the packages.

Eventually, with the approval from substack, Jordan was able to get complete ownership of over a dozen of the impacted packages.

Since Jordan is already a Tidelift partnered maintainer, paid by Tidelift to implement enterprise class secure software development practices across many of the JavaScript packages he maintains, he was able to add minimist to the list of packages he receives income for from Tidelift.

And because this income is made possible by Tidelift customers (Tidelift pays maintainers based on factors like customer usage and package criticality), Jordan has already documented many of the important secure development practices followed by minimist, including providing vulnerability fixes for the latest release, monitoring dependencies for issues, having 2FA enabled, having a secure vulnerability disclosure process, and more.

Even better, Jordan was able to bring in an experienced co-maintainer for minimist, who also shares in the revenue from Tidelift and provides extra support to ensure that it is no longer a solo maintainer project like it was before, making the future resilience for minimist even greater.

Now, Tidelift customers can use minimist—and other packages that rely on it—with confidence, knowing that two experienced maintainers have made the commitment to ensure the package follows enterprise secure software development practices. What's more, Tidelift customers directly played a role in funding work on minimist so that Jordan can make investments in ensuring it stays resilient and healthy into the future.