Firefox

# Package manager security

or, why you should be scared of npm (if you’re not already)

**08.18.20**

alex bellon — security engineering intern

# A little bit about me

Rising senior at UT Austin studying math and computer science (focus in security)

Security Engineering Intern with Firefox Operations Security

Project focused on how package dependencies can affect the security and integrity of a codebase

# Motivation

Attacks on packages by way of compromising dependencies are becoming more and more common

- `leftpad` - a maintainer unpublished their package on NPM, and another user republished a copy of the package under the same name. Luckily the user was well-meaning, but it could have gone the other way, and nobody would have noticed (Mar 2016)
- `eslint` - a maintainer’s account was compromised and a malicious release was published (July 2018)
- `event-stream` - attacker got added as a maintainer and published malicious code (Nov 2018)

# Motivation

Mozilla repositories and services have vulnerabilities, whether we know about them or not

I found multiple Mozilla repositories that use packages with active vulnerabilities and/or packages that are unmaintained

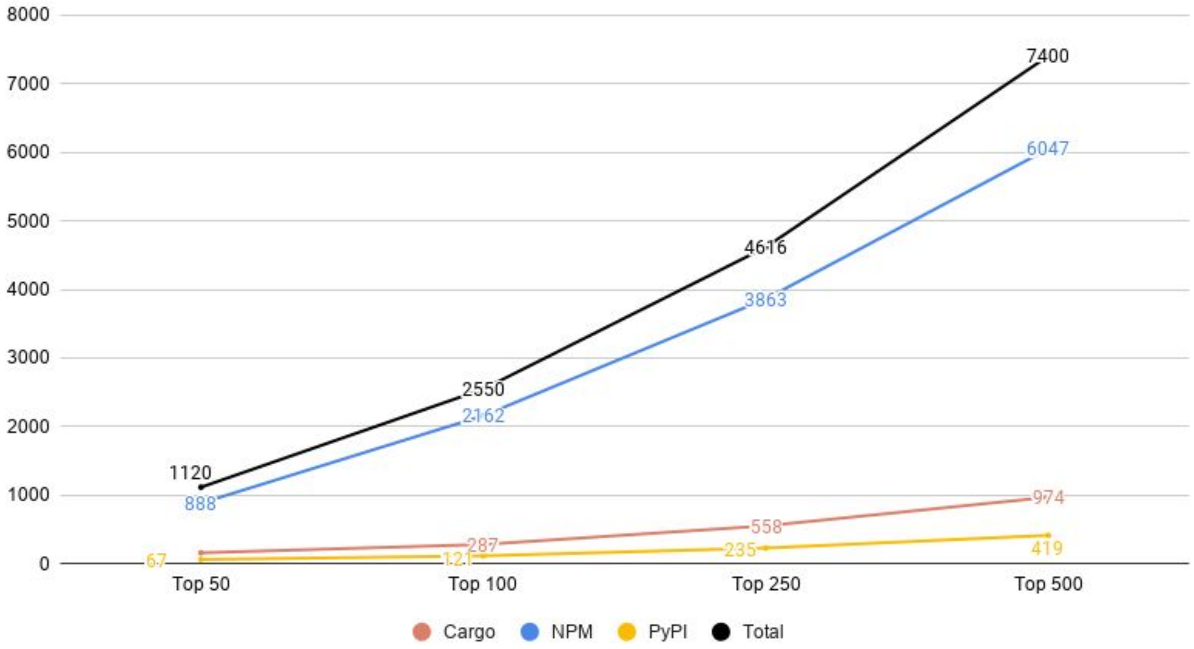
...and that only accounts for the vulnerabilities that have been found

# Initial research

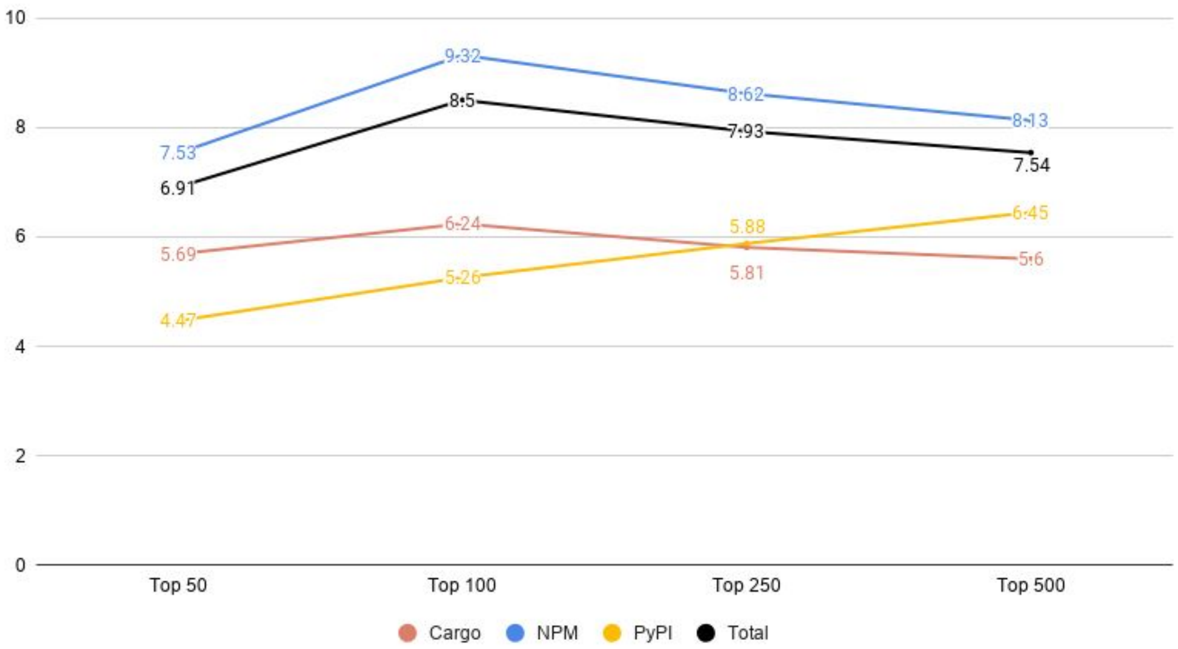
For the first few weeks of my internship, I looked at the status of different package managers and ecosystems

I specifically focused on NPM (Node/JS), Cargo (Rust) and PyPI (Python) to see how vulnerable top packages were to attack

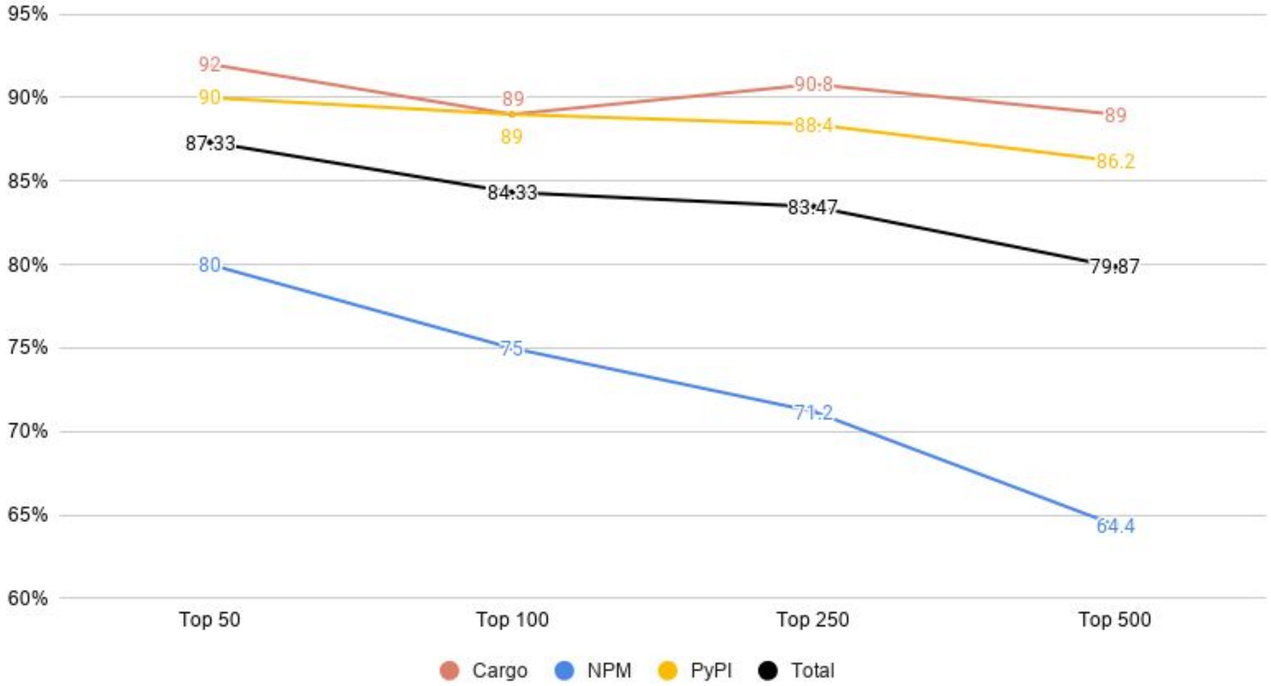
Two things I looked at were the number of leaks that package maintainers/contributors had on HaveIBeenPwned (HIBP) and the number of maintained packages (release within the past year)



## Total number of HIBP leaks for all maintainers/contributors for the top X packages in an ecosystem



**Average number of HIBP leaks per person for the top X packages in an ecosystem**

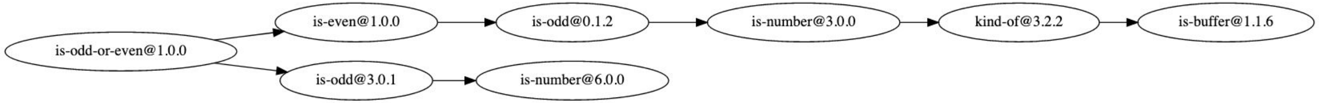


# Percentage of packages that are actively maintained for the top X packages in an ecosystem

# Takeaways

The design and culture of NPM’s ecosystem (micropackages, many dependencies) leads to packages that have lots of attack vectors

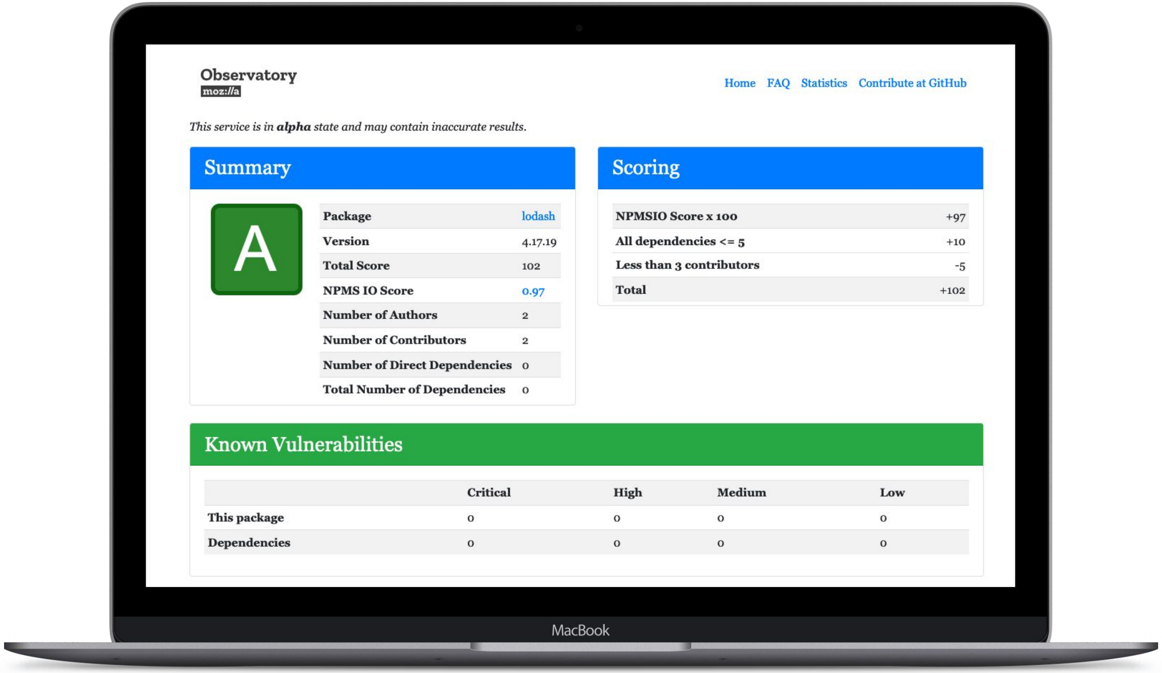
This means that using NPM packages introduces a large attack surface into your code that you may not be aware or in control of



The package `is-odd-or-even`, which (you guessed it) tells you if a number is odd or even, has 7 dependencies! 2 of them are just different versions of packages that are already included!!

# Dependency Observatory

Dependency Observatory allows developers to see how secure a package is before adding it as a dependency to their project



Observatory
moz://a

Home FAQ Statistics Contribute at GitHub

*This service is in **alpha** state and may contain inaccurate results.*

## Summary

A

| | |
|---|---|
| Package | lodash |
| Version | 4.17.19 |
| Total Score | 102 |
| NPMS IO Score | 0.97 |
| Number of Authors | 2 |
| Number of Contributors | 2 |
| Number of Direct Dependencies | 0 |
| Total Number of Dependencies | 0 |

## Scoring

| | |
|---|---|
| NPMSIO Score x 100 | +97 |
| All dependencies <= 5 | +10 |
| Less than 3 contributors | -5 |
| Total | +102 |

## Known Vulnerabilities

| | Critical | High | Medium | Low |
|---|---|---|---|---|
| This package | 0 | 0 | 0 | 0 |
| Dependencies | 0 | 0 | 0 | 0 |

https://stage.depobs.nonprod.cloudops.mozgcp.net

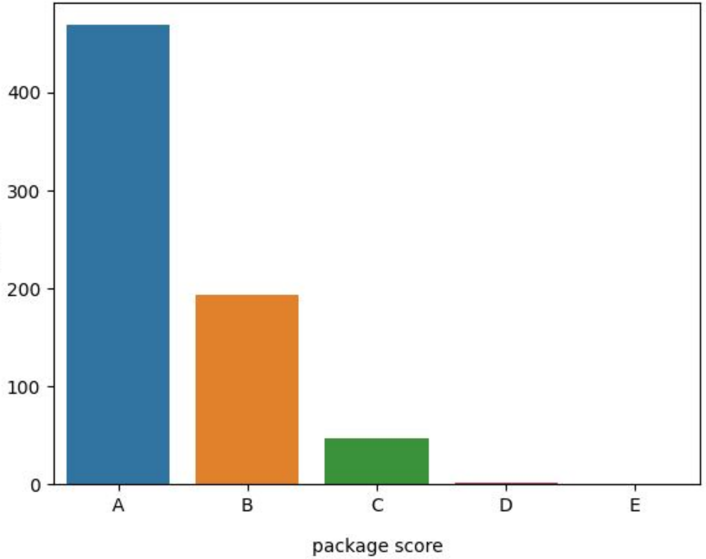
# Dependency Observatory

Using what I learned from my research, I was able to change the scoring algorithm used to rate the security of packages to more accurately reflect how secure they were

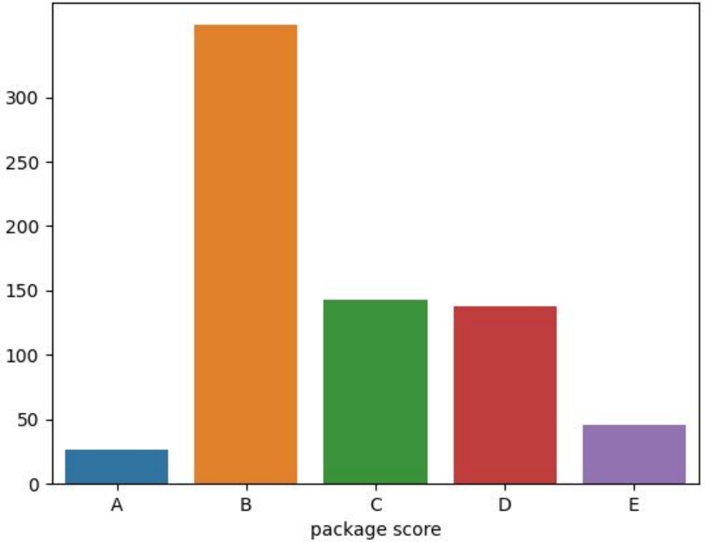
- Weigh # vulnerabilities more than # maintainers (used to be able to cancel each other out)
- Remove points per vulnerability, not just once
- Packages with high/critical vulns cannot score an A
- Remove points for a large number of maintainers

# Dependency Observatory

before algorithm changes



after algorithm changes



# Dependency Observatory

Added a CLI command to directly load advisories from the GitHub Advisory Database into the project database

Added a CLI command to load the HIBP breaches for all maintainers/contributors of a package into the project database

Added a feature that allows you to compare the dependency trees of two packages and get the diffs of the dependencies

Added a statistics page with graphs that represent the distribution of package scores

Plus some other miscellaneous bugs

# Thank you!

Greg Guthe (mentor)

Julien Vehent (manager)

FoxSec team

University team

All the interns (especially Internal Flame)