Small World with High Risks: A Study of Security Threats in the npm Ecosystem

Markus Zimmermann - TU Darmstadt
Cristian-Alexandru Staicu - TU Darmstadt
Cam Tenny - r2c
Michael Pradel - TU Darmstadt

August 15, 2019

TECHNISCHE UNIVERSITÄT DARMSTADT

SOLA SoftwareLab

JavaScript and npm

Most Popular Technologies

Programming, Scripting, and Markup Languages

All Respondents

JavaScript	69.8%
HTML	68.5%
CSS	65.1%
SQL	57.0%
Java	45.3%
Bash/Shell	39.8%
Python	38.8%
C#	34.4%

https://insights.stackoverflow.com/survey/2018/

By the numbers

Packages

697,156

Downloads · Last Day

870,713,527

Downloads · Last Week

4,746,761,893

Downloads · Last Month

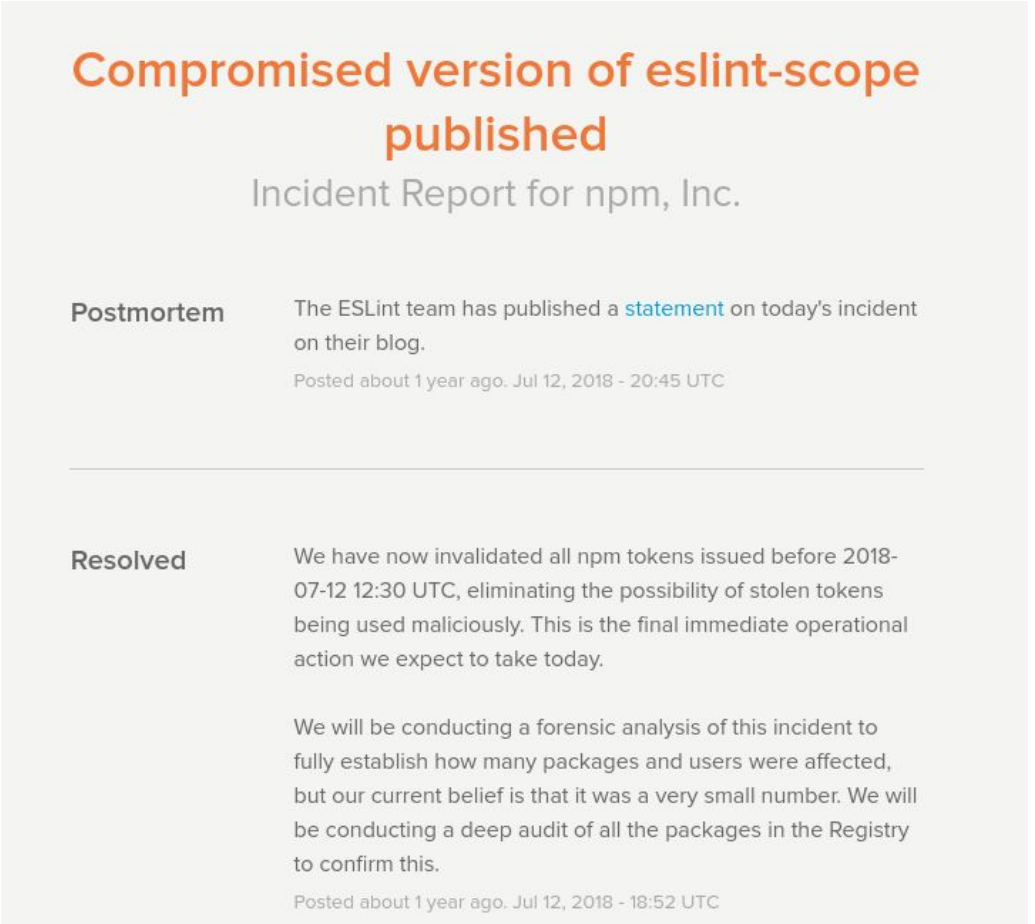
20,522,084,656

npmjs.com retrieved on May 8, 2018

609 security advisories

eslint Incident

- ~10 million downloads
- Compromised credentials of maintainer
- Malicious version tried to steal access tokens



Compromised version of eslint-scope published

Incident Report for npm, Inc.

Postmortem

The ESLint team has published a statement on today's incident on their blog.

Posted about 1 year ago. Jul 12, 2018 - 20:45 UTC

Resolved

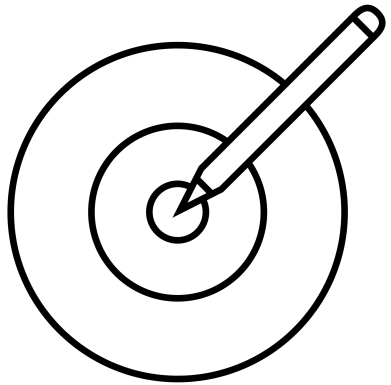
We have now invalidated all npm tokens issued before 2018-07-12 12:30 UTC, eliminating the possibility of stolen tokens being used maliciously. This is the final immediate operational action we expect to take today.

We will be conducting a forensic analysis of this incident to fully establish how many packages and users were affected, but our current belief is that it was a very small number. We will be conducting a deep audit of all the packages in the Registry to confirm this.

Posted about 1 year ago. Jul 12, 2018 - 18:52 UTC

Goals

- What is the impact of an individual package on the ecosystem?
- What is the influence of a maintainer on the ecosystem?
- How many packages depend on a package with unpatched security vulnerabilities?



Key Findings

- On average, packages implicitly trust **79** third-party packages and **39** maintainers
- Popular packages often influence **more than 100,000** other packages
- Some maintainers have an impact on **hundreds of thousands** of packages
- Up to **40%** of all packages depend on code with at least one publicly known vulnerability

Particularities of npm

- No vetting of developers
- Open publishing model
- Heavy reuse
- Locked dependencies

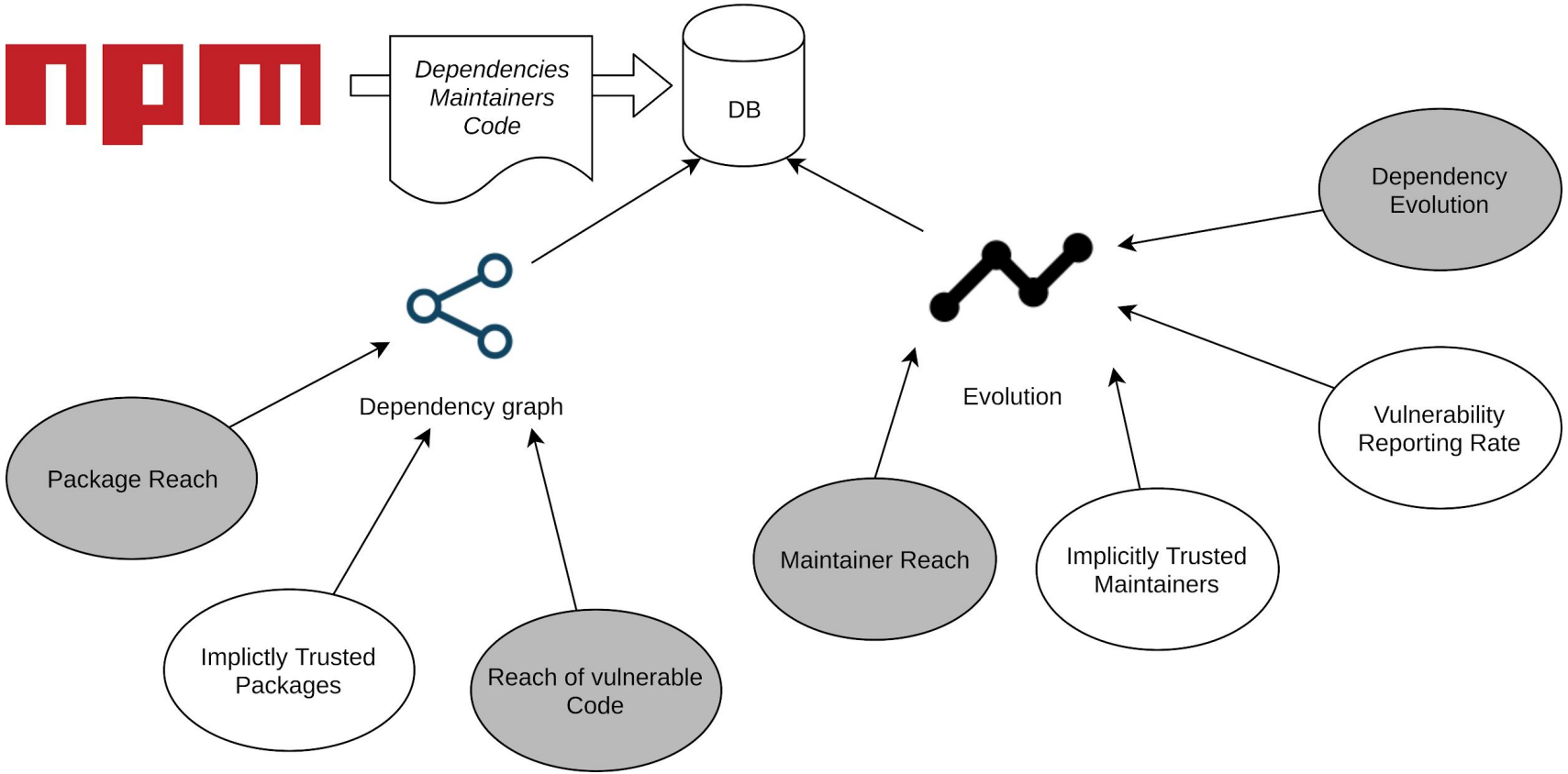
Malicious packages

Package / Account takeover

Maintainer collusion

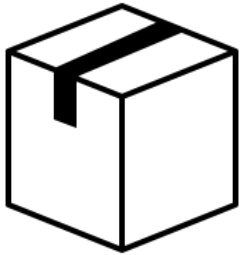
Exploiting unmaintained legacy code

Empirical Study

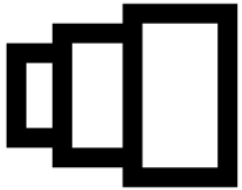


Experimental Setup

Observation period: **November 2010 - April 2018**



676,539 packages

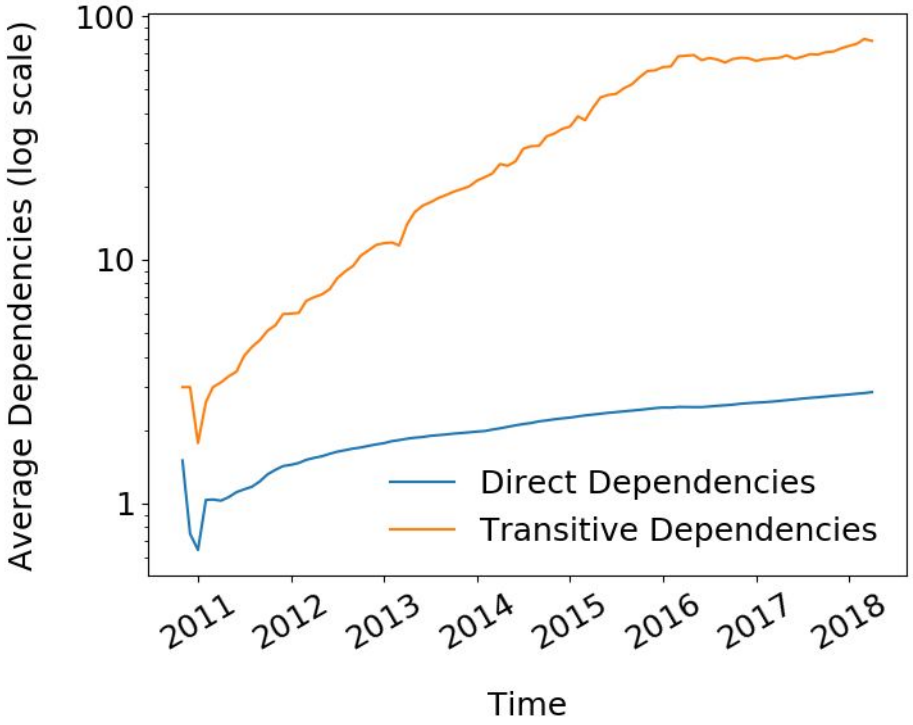


5,386,239 versions

199,327 maintainers

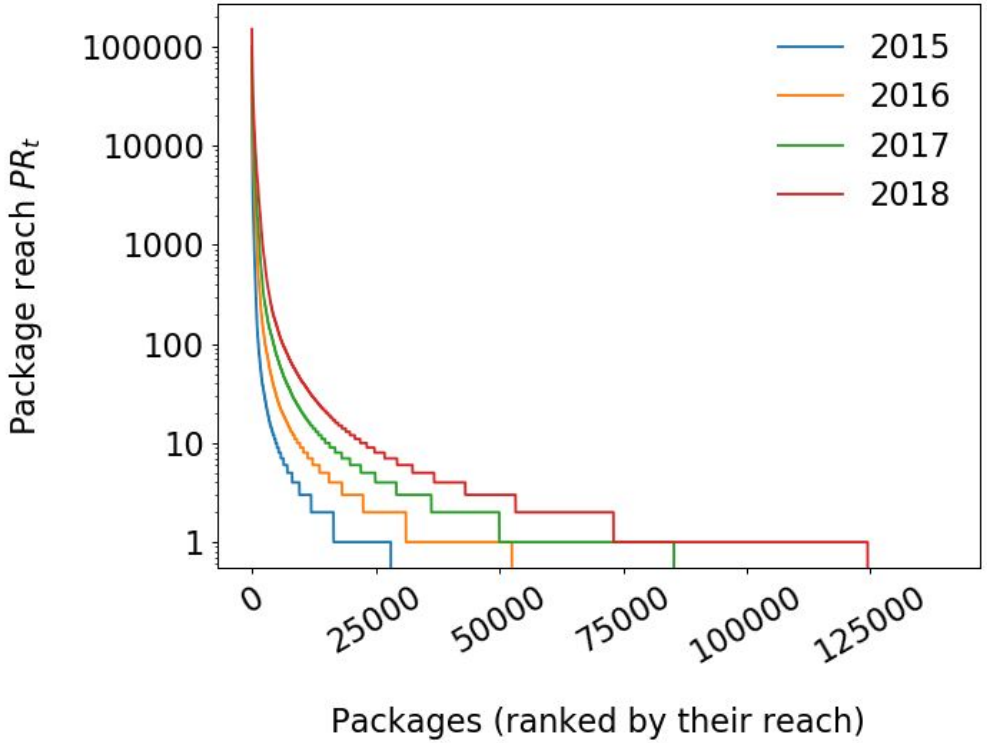
609 security advisories

Evolution of Dependencies



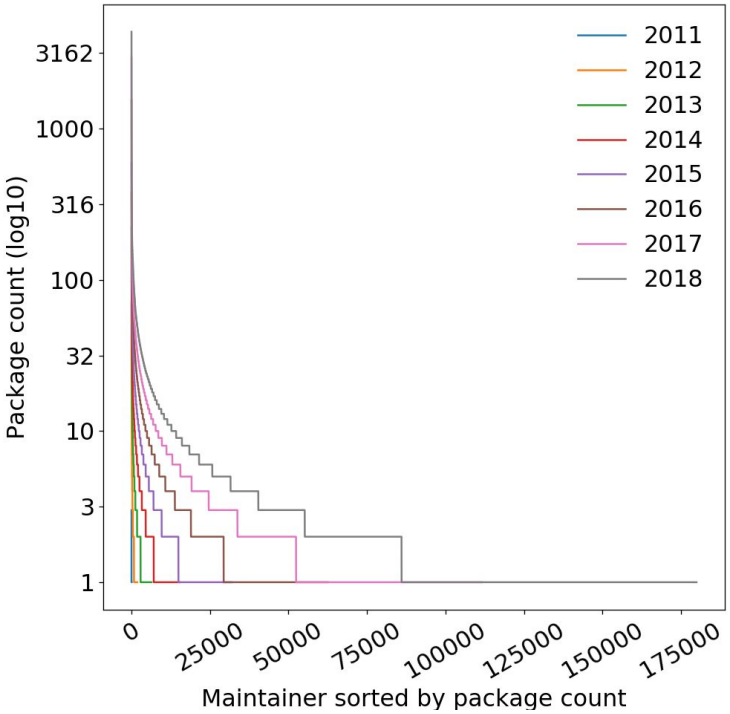
A user implicitly trusts 79 other packages due to transitive dependencies.

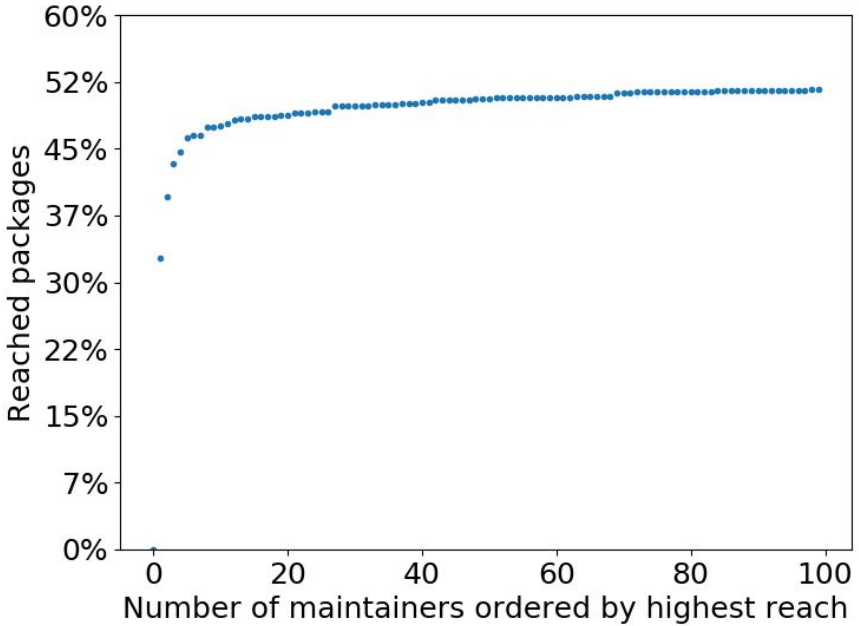
Evolution of Package Reach



Popular packages can reach more than 100,000 other packages, making them a prime target for attacks.

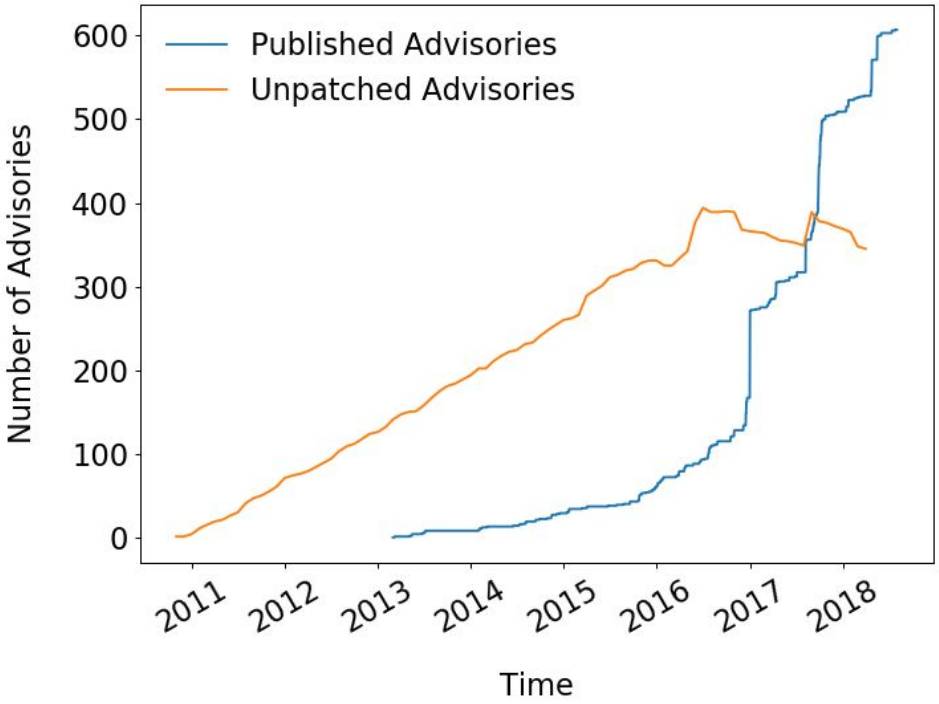
Evolution of Maintainer Influence

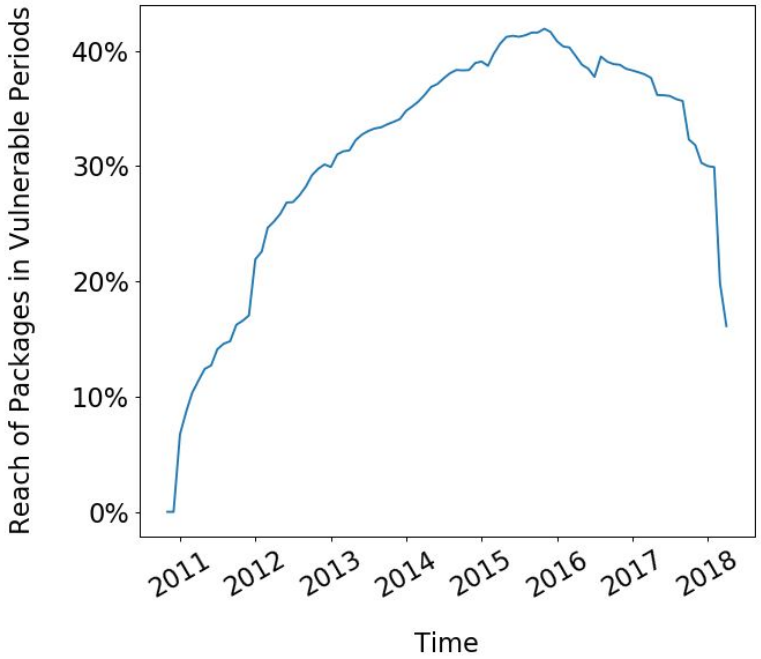




Less than 100 maintainers are needed to reach 50%,
22.2 % are needed to reach all with dependencies

Evolution of Security Advisories



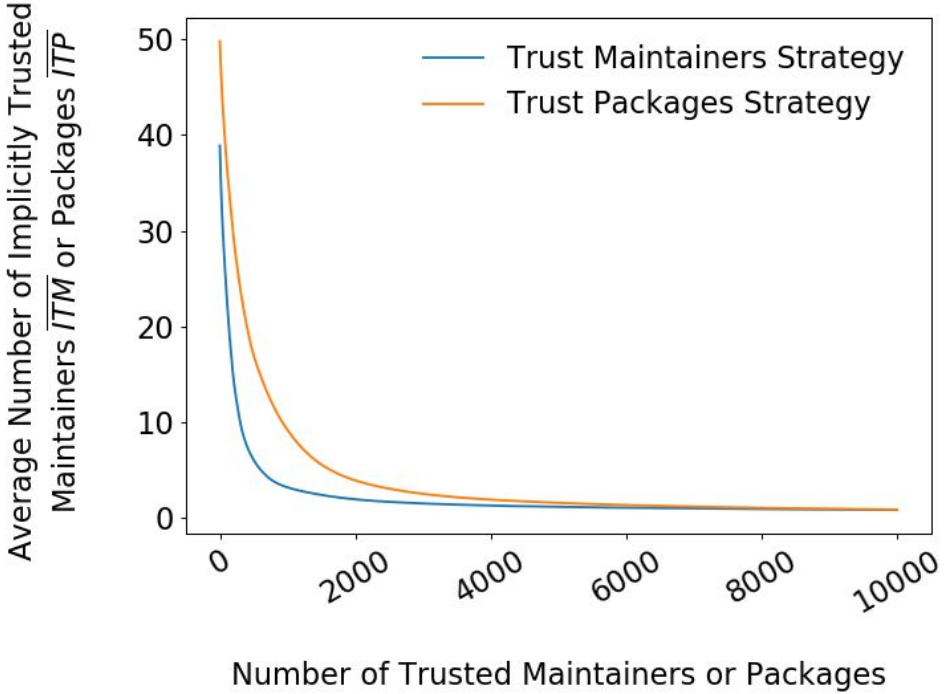


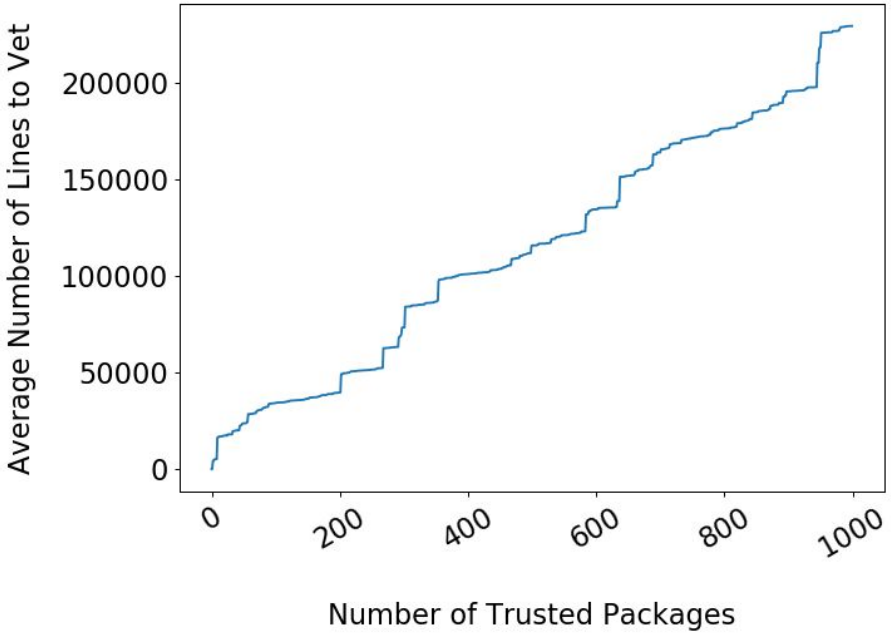
Up to 40% of all packages rely on code known to be vulnerable.

Potential Mitigations

- Raising developer awareness of dependency risks
- Warn about vulnerable packages
- Vetting code or maintainers

Code Vetting as Mitigation





Vetting the most dependent upon 1,500 packages would reduce the implicitly trusted packages by a factor of 10

Conclusions

- On average, packages implicitly trust **79** third-party packages and **39** maintainers
- Popular packages influence often **more than 100,000** other packages
- Some maintainers have an impact on **hundreds of thousands** of packages
- Up to **40%** of all packages depend on code with at least one publicly known vulnerability