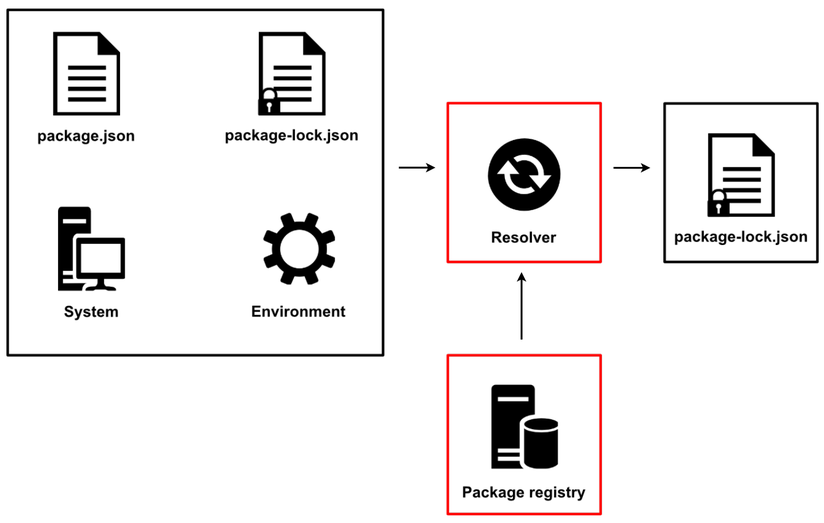
ReSolVer
Resolution Integrity for npm

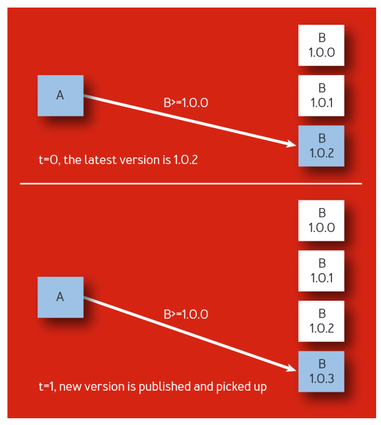
Timothy Lindquist

Zero Trust

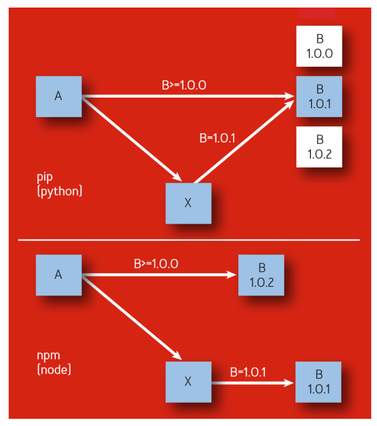


The Surprise of Multiple Dependency Graphs

Registry state changes over time as new packages and versions are published



Different resolvers produce different dependency graphs



Figures from Anugerah & Martin-Jones [1]

Record

Integrity Provenance

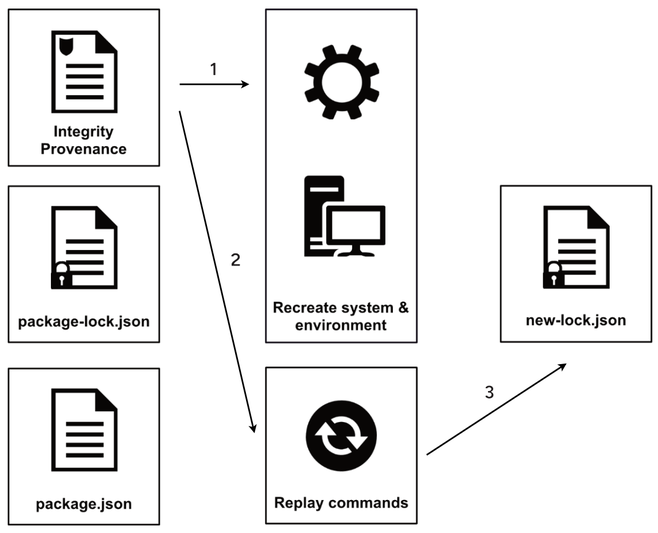
lockfileDigest

systemInfo

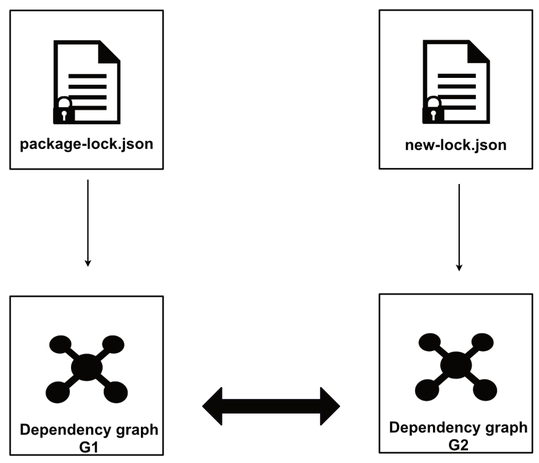
environmentInfo

commands:
[
[command1, timestamp1],
[command2, timestamp2],
...
]

ReSolve



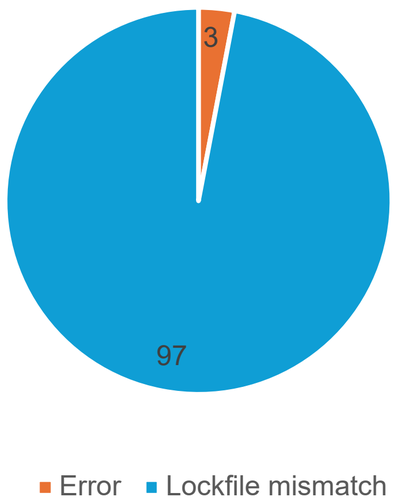
Verify



Preliminary Results

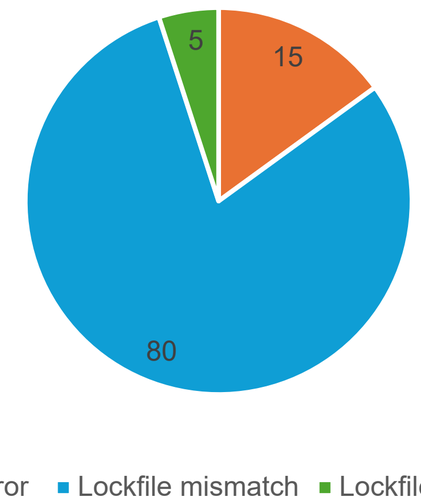
Three experiments were performed on 100 projects that published both a package.json and package-lock.json

Experiment 1



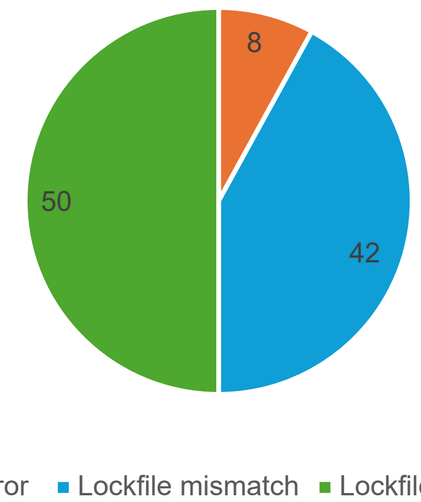
This experiment used the published manifest and ran the command `npm install`.

Experiment 2



This experiment used the published manifest and ran the command `npm install` but filtered out packages that were published after the commit time.

Experiment 3



This experiment used a manifest and lockfile from a previous commit. These were used to infer commands and resolution time between the two commits. The commands were then executed with package filtering according to the inferred time.

WARNING: Do not attempt these experiments at home. If you do, use the `--ignore-scripts` flag.
For more information, see any cybersecurity news site, any day of the week.

[1] Anugerah, Josie, and Eve Martin-Jones. 2025. "The Surprise of Multiple Dependency Graphs: Dependency Resolution Is Not Deterministic." Queue 23(1): 64–84. doi:10.1145/3723000.