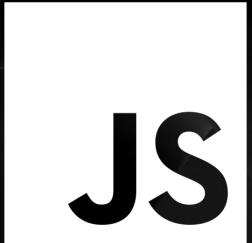
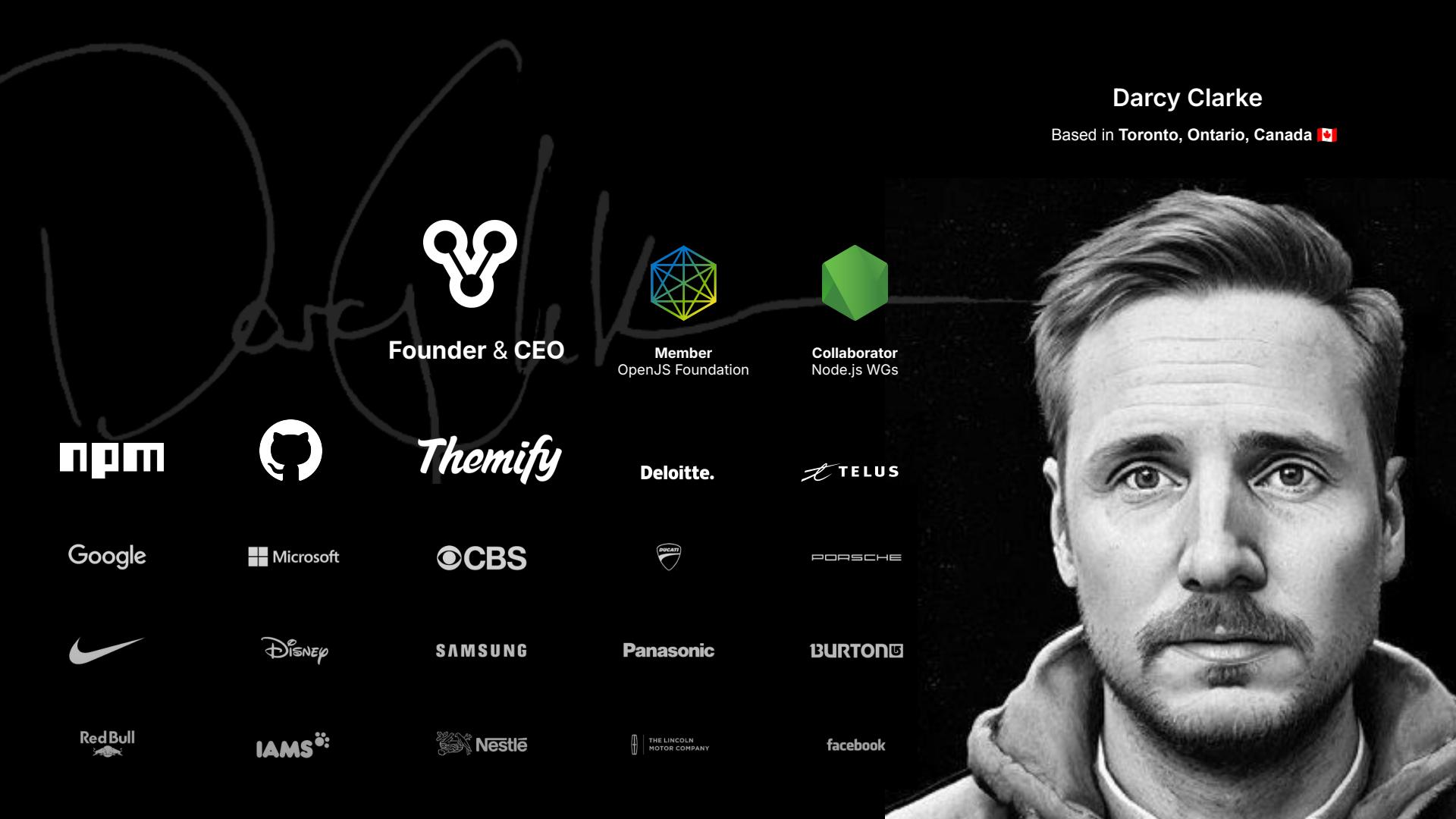


Modernizing Supply Chain Security



 tinyurl.com/modern-2025





Darcy Clarke

Based in Toronto, Ontario, Canada 



Founder & CEO



Member
OpenJS Foundation



Collaborator
Node.js WGs



Themify

Deloitte.

TELUS

Google

Microsoft

CBS



PORSCHE



Disney

SAMSUNG

Panasonic

BURTON

RedBull

IAMS

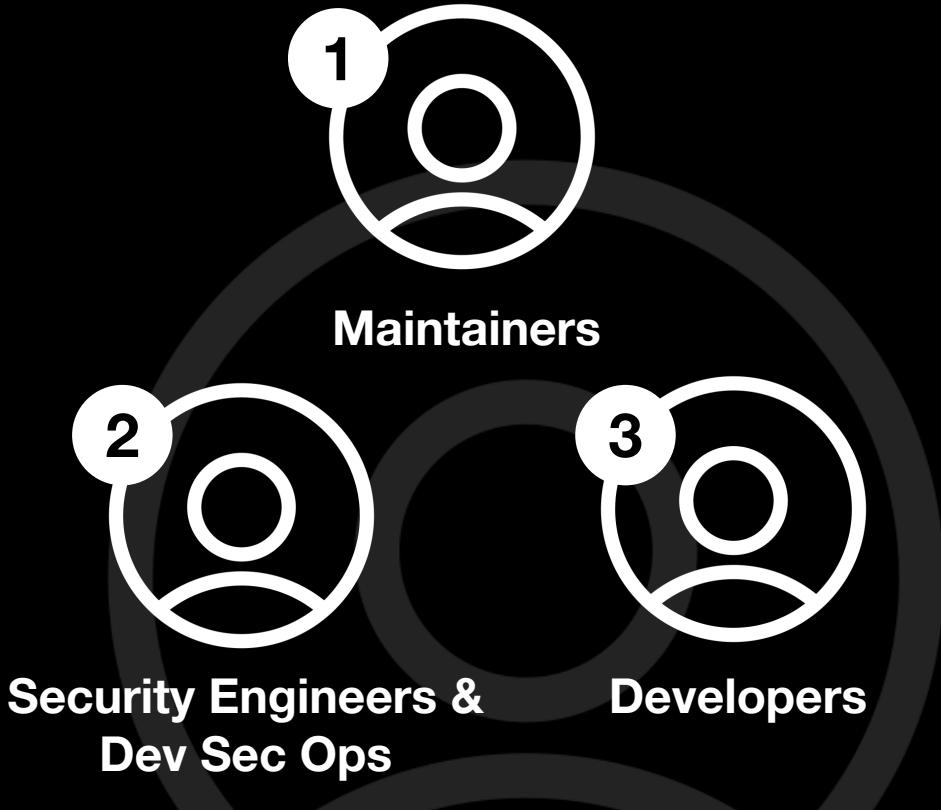
Nestlé

THE LINCOLN
MOTOR COMPANY

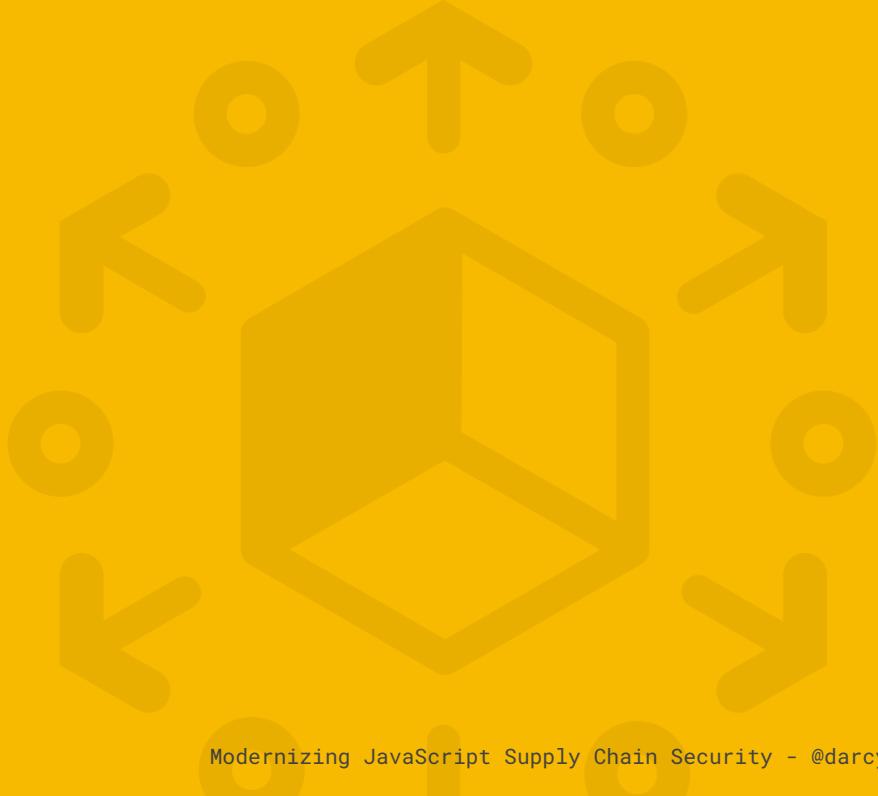
facebook



Who Are you?



Current State **Ecosystem**



JS**Runtimes****Runtime Managers****Bundlers****CI/CD Automation & Security****Private Registries****Monorepos & Build Pipelines****Package Managers****Public Registries****Open Source Registries**

JS**Runtimes****Runtime Managers****Bundlers****CI/CD Automation & Security****Private Registries****Monorepos & Build Pipelines****Package Managers****Public Registries****Open Source Registries**

CVEs - Common Vulnerabilities & Exposures

Screenshot of the GitHub Advisory Database interface.

The page title is "GitHub Advisory Database". The URL in the address bar is "github.com/advisories".

The main heading is "GitHub Advisory Database". Below it is a sub-heading: "Security vulnerability database inclusive of CVEs and GitHub originated security advisories from the world of open source software."

A sidebar on the left lists "GitHub reviewed advisories" with the following counts:

Category	Count
All reviewed	21,400
Composer	4,386
Erlang	33
GitHub Actions	22
Go	2,141
Maven	5,315
npm	3,803
NuGet	687
pip	3,480
Pub	12
RubyGems	897
Rust	898
Swift	38

A search bar at the top right says "Search by CVE/GHSA ID, package, severity, ecosystem, credit..."

The main content area shows a list of vulnerabilities:

21,400 advisories		Severity ▾	CWE ▾	Sort ▾
Mautic allows Relative Path Traversal in assets file upload Moderate				
CVE-2022-25773 was published for mautic/core (Composer) 15 hours ago				
Mautic allows Improper Authorization in Reporting API High				
CVE-2024-47053 was published for mautic/core (Composer) 15 hours ago				
Mautic allows Remote Code Execution and File Deletion in Asset Uploads Critical				
CVE-2024-47051 was published for mautic/core (Composer) 15 hours ago				
copyparty renders unsanitized filenames as HTML when user uploads empty files Low				
CVE-2025-27145 was published for copyparty (pip) 15 hours ago				
io.quarkus:quarkus-resteasy: Memory Leak in Quarkus RESTEasy Classic When Client Requests Timeout High				
CVE-2025-1634 was published for io.quarkus:quarkus-resteasy (Maven) 17 hours ago				
Matrix IRC Bridge allows IRC command injection to own puppeted user Low				
CVE-2025-27146 was published for matrix-appservice-irc (npm) 2 days ago				
DOM Expressions has a Cross-Site Scripting (XSS) vulnerability due to improper use of string.replace High				
CVE-2025-27108 was published for dom-expressions (npm) 2 days ago				
Solid Lacks Escaping of HTML in JSX Fragments allows for Cross-Site Scripting (XSS) High				
CVE-2025-27109 was published for solid-js (npm) 2 days ago				

```
~/desktop/test/test-pkg (0.888s)
```

```
npm i
```

```
removed 2 packages, and audited 122 packages in 783ms
```

```
14 packages are looking for funding  
  run `npm fund` for details
```

9 vulnerabilities (2 low, 3 moderate, 3 high, 1 critical)

```
To address all issues possible (including breaking changes), run:  
  npm audit fix --force
```

```
Some issues need review, and may require choosing  
a different dependency.
```

```
Run `npm audit` for details.
```

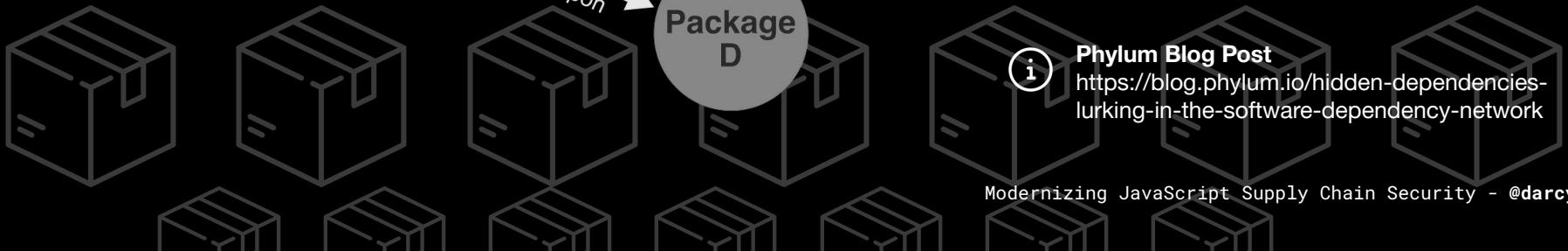
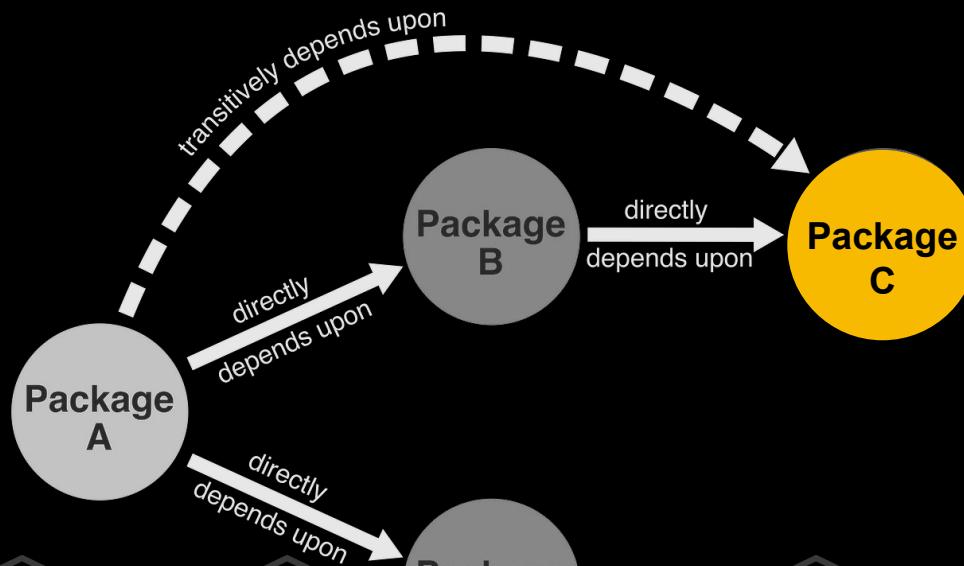


It's estimated **75%**
of vulnerabilities reside in
transitive dependencies



Snyk's State of Open Source Security 2020:
<https://snyk.io/series/open-source-security/report-2020/>

Transitive Dependencies



Phylum Blog Post
<https://blog.phylum.io/hidden-dependencies-lurking-in-the-software-dependency-network>

742%

year-over-year increase in attacks
targeting the open source supply chain



Sonatype's State of Software Supply Chain Security 2022:
<https://www.sonatype.com/state-of-the-software-supply-chain/>

What are they

Exploiting?

- **Vulnerabilities**
- **Typosquatting**
- **Dependency/Manifest Confusion**
- **Registry Compromise**
- **Account Takeovers**
- **Malware**

How can we mitigate

Vulnerabilities

- **CVEs**
- **Active Scanning Tools**
- **Write better code**

Example: create-react-app

```
→ test-react git:(main) ✘ npm audit
# npm audit report

nth-check <2.0.1
Severity: high
Inefficient Regular Expression Complexity in nth-check - https://github.com/advisories/GHSA-rp65-9cf3-cjxr
fix available via `npm audit fix --force`
Will install react-scripts@2.1.3, which is a breaking change
node_modules/svgo/node_modules/nth-check

css-select <=3.1.0
Depends on vulnerable versions of nth-check
node_modules/svgo/node_modules/css-select

svgo 1.0.0 - 1.3.2
Depends on vulnerable versions of css-select
node_modules/svgo

@svgr/plugin-svgo <=5.5.0
Depends on vulnerable versions of svgo
node_modules/@svgr/plugin-svgo

@svgr/webpack 4.0.0 - 5.5.0
Depends on vulnerable versions of @svgr/plugin-svgo
node_modules/@svgr/webpack

react-scripts >=2.1.4
Depends on vulnerable versions of @svgr/webpack
node_modules/react-scripts
```

6 **high** severity vulnerabilities

To address all issues (including breaking changes), run:

```
npm audit fix --force
```



I DON'T KNOW HOW TO USE NPM
INSTALL WITHOUT GETTING 104 VULNERABILITIES

AND AT THIS POINT
I'M TOO AFRAID TO ASK

Example: create-react-app

```
→ test-react git:(main) ✘ npm audit
# npm audit report

nth-check <2.0.1
Severity: high
Inefficient Regular Expression Complexity in nth-check - https://github.com/advisories/GHSA-rp65-9cf3-cjxr
fix available via `npm audit fix --force`
Will install react-scripts@2.1.3, which is a breaking change
node_modules/svgo/node_modules/nth-check
  css-select <=3.1.0
    Depends on vulnerable versions of nth-check
    node_modules/svgo/node_modules/css-select
      svgo 1.0.0 - 1.3.2
        Depends on vulnerable versions of css-select
        node_modules/svgo
          @svgr/plugin-svgo <=5.5.0
            Depends on vulnerable versions of svgo
            node_modules/@svgr/plugin-svgo
          @svgr/webpack 4.0.0 - 5.5.0
            Depends on vulnerable versions of @svgr/plugin-svgo
            node_modules/@svgr/webpack
          react-scripts >=2.1.4
            Depends on vulnerable versions of @svgr/webpack
            node_modules/react-scripts
```

6 **high** severity vulnerabilities

To address all issues (including breaking changes), run:
npm audit fix --force



Example: create-react-app

```
npm audit --fix
→ test-react git:(main) ✘ npm audit --fix
( ) :: audit: timing arborist:ctor Compl
```



Severity: **high**

Terser insecure use of regular expressions leads to ReDoS - <https://github.com/advisories/GHSA-4wf5-vphf-c2xc>

fix available via `npm audit fix --force`

Will install react-scripts@5.0.1, which is a breaking change

node_modules/terser

yargs-parser 6.0.0 - 13.1.1

Severity: **moderate**

yargs-parser Vulnerable to Prototype Pollution - <https://github.com/advisories/GHSA-p9pc-299p-vxgp>

fix available via `npm audit fix --force`

Will install react-scripts@5.0.1, which is a breaking change

node_modules/webpack-dev-server/node_modules/yargs-parser

node_modules/yargs-parser

yargs 8.0.0-candidate.0 - 12.0.5

Depends on vulnerable versions of **yargs-parser**

node_modules/webpack-dev-server/node_modules/yargs

node_modules/yargs

84 vulnerabilities (14 low, 19 moderate, 43 high, 8 critical)

To address issues that do not require attention, run:

npm audit fix

To address all issues (including breaking changes), run:

npm audit fix --force



Example: create-react-app

Modernizing JavaScript Supply Chain Security - @darcy



How can we mitigate

Typosquatting

- **Heuristics**
(name, downloads, versions, published date, author etc.)
- **Policies & Enforcement**

How can we mitigate

Dependency Confusion

- Use a publicly owned scope for internal/private proxied packages
- Set registry configuration in a `.npmrc` file at the root-level of your projects
- Respond quickly to build failures
- Introduce per-package registry protocol to package specifier



Avoiding npm substitution attacks

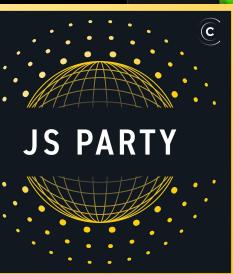
<https://github.blog/2021-02-12-avoiding-npm-substitution-attacks/>

How can we mitigate

Manifest Confusion

- **Do not respect registry manifest data that is authored**
- **Use a package's extracted package.json as the source of truth**

"is npm broken by design?"



<https://blog.vlt.sh/blog/the-massive-hole-in-the-npm-ecosystem>

The screenshot shows a blog post titled "The massive bug at the heart of the npm ecosystem" by Darcy Clarke. The post features a vibrant, abstract background with various npm package names like "sinatra", "socket.io", "optimist", and "phoenix" visible through a distorted, colorful grid. At the top left, there's a cartoon skull and crossbones icon. The author's profile picture and name are at the bottom left, along with a disclosure note about their past role at npm. The bottom right corner has the word "tldr;".

The massive bug at the heart of the npm ecosystem

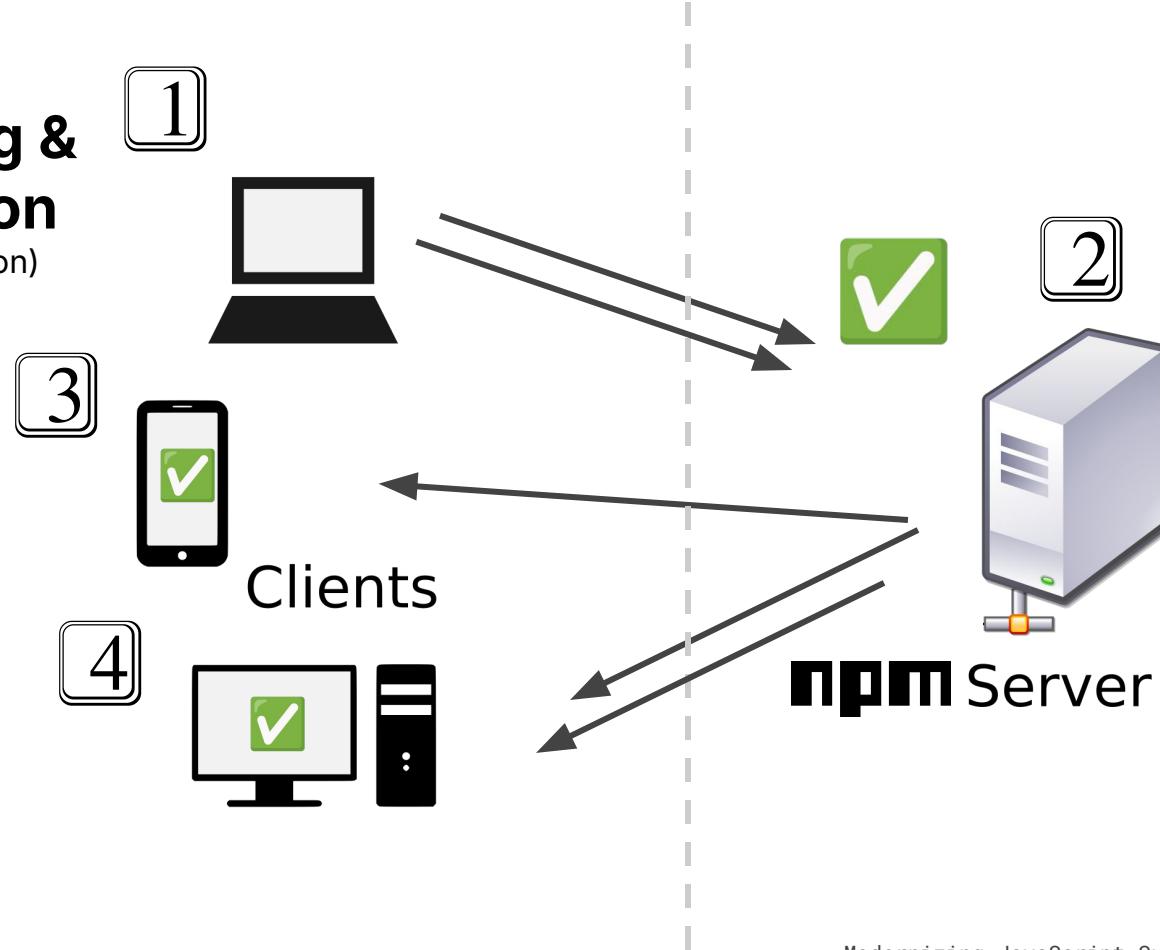
Darcy Clarke

Disclosure: I was the Staff Engineering Manager for the npm CLI team between July 2019 & December 2022. I was a part of the GitHub acquisition of npm inc. in 2020. I left GitHub, for various reasons, in December.

tldr;

Package Publishing & Distribution

(Manifest Confusion)

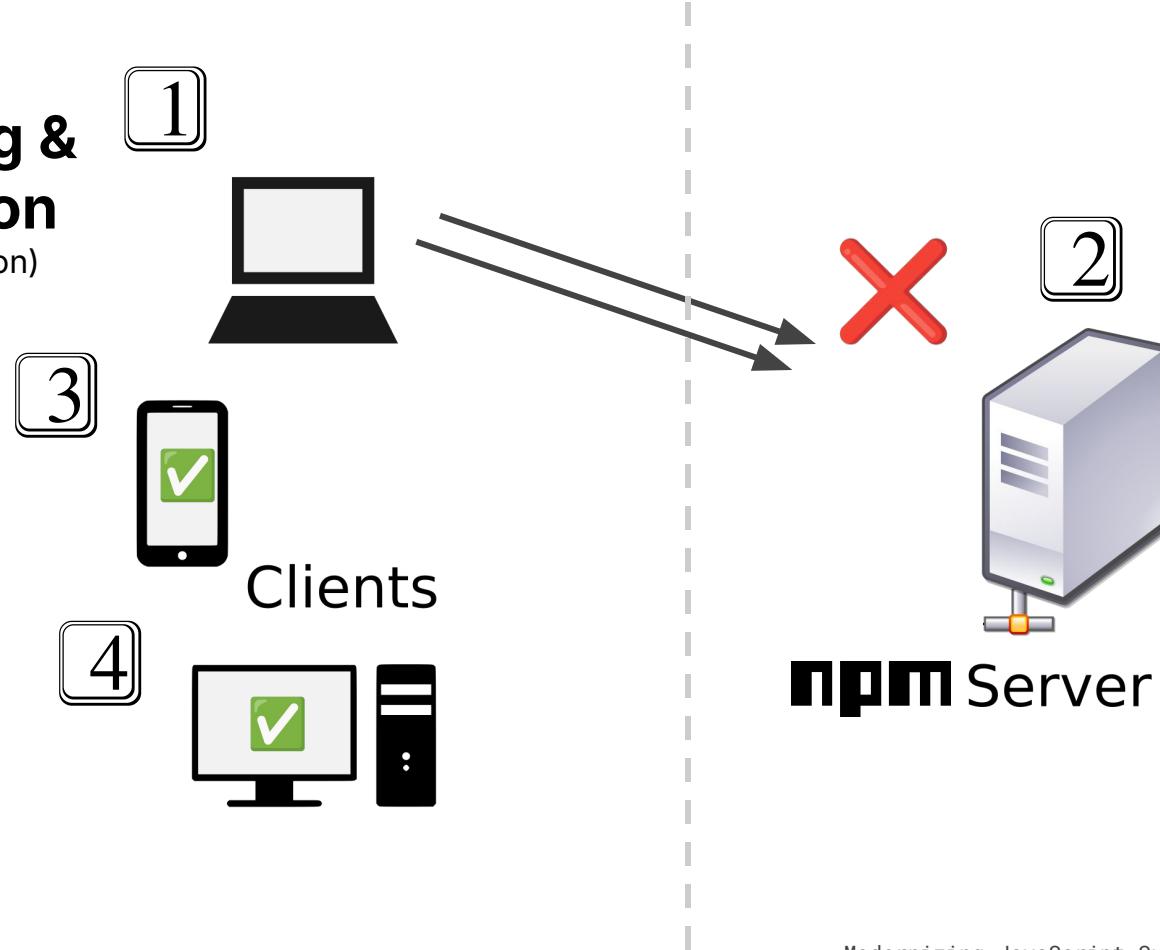




Modernizing JavaScript Supply Chain Security - @darcy

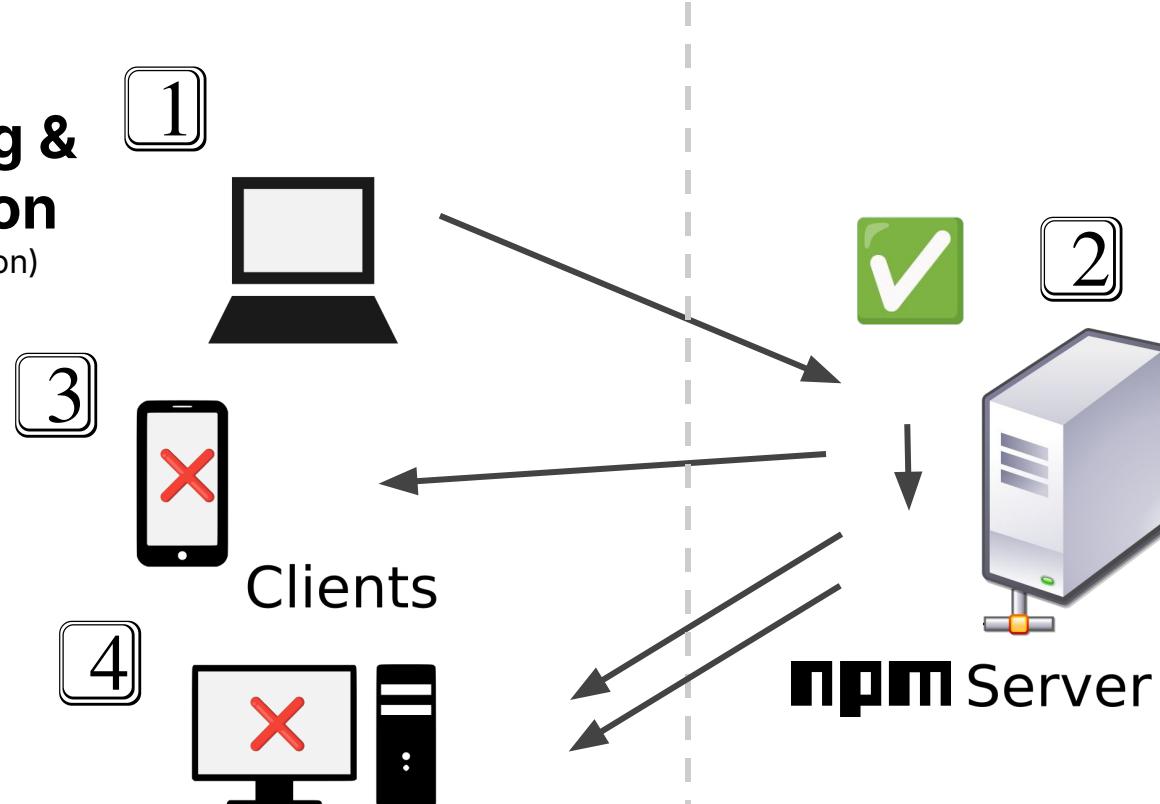
Package Publishing & Distribution

(Manifest Confusion)



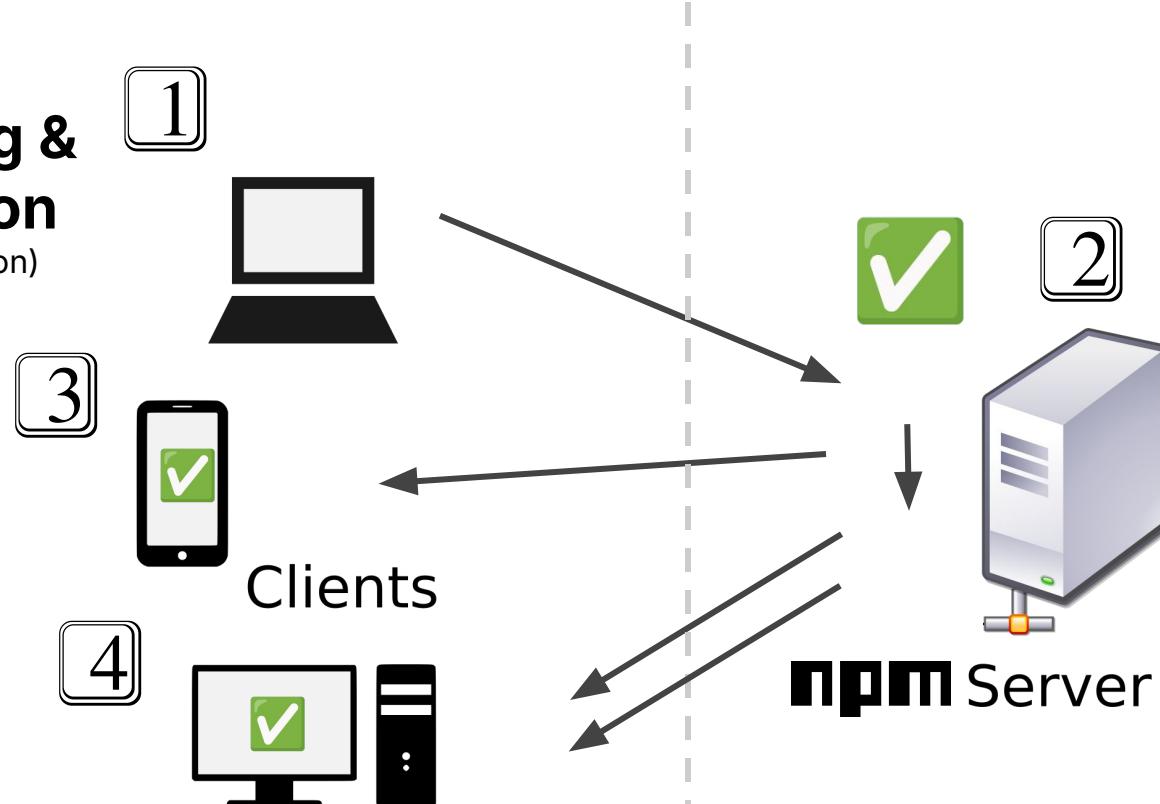
Package Publishing & Distribution

(Manifest Confusion)



Package Publishing & Distribution

(Manifest Confusion)





socket.dev

npm - npm Package Overview

Big news! Introducing Socket AI - ChatGPT-Powered Threat Analysis.

npm package overview for darcyclarke-manifest-pkg

npm 4.0.2

Package Overview

Dependencies 85

Maintainers 4

Versions 521

Issues

File Explorer

ADVANCED TOOLS

npm Scripts

CVE VULNERABILITY Contains a high severity Common Vulnerability and Exposure (CVE). Found 4 instances in 1 package

Bin script shell injection SUPPLY CHAIN RISK This package re-exports a well known shell command via an npm bin. Found 1 instance in 1 package

Network access SUPPLY CHAIN RISK This module accesses the network. Found 8 instances in 1 package

Shell access SUPPLY CHAIN RISK This module accesses the system shell. Accessing the system shell increases the risk that the code may contain exploits or malicious behavior. Found 1 instance in 1 package

Uses eval SUPPLY CHAIN RISK Package uses eval() which is a dangerous function. This prevents the code from being safely evaluated. Found 1 instance in 1 package

Version published 7 years ago

Maintainers 4

Weekly downloads 4,700,483 ▲ 8.05%

Readme

darcyclarke-manifest-pkg - npm

Big news! Introducing Socket AI - ChatGPT-Powered Threat Analysis. Learn more →

Socket Product Resources Docs Pricing npm Search for npm package Log in Demo Install

DARCYCLARKE-MANIFEST-PKG 2.1.15 (latest)

Supply Chain Security Quality Maintenance Vulnerabilities License

Manifest confusion SUPPLY CHAIN RISK

This package has inconsistent metadata. This could be malicious or caused by an error when publishing the package. Found 1 instance in 1 package

Install scripts SUPPLY CHAIN RISK

Install scripts are run when the package is installed. The majority of malware in npm is hidden in install scripts. Found 1 instance in 1 package

Floating dependency QUALITY

Package has a dependency with a floating version range. This can cause issues if the dependency publishes a new major version. Found 1 instance in 1 package

No README QUALITY

Package does not have a README. This may indicate a failed publish or a low quality package. Found 1 instance in 1 package

Trivial Package SUPPLY CHAIN RISK

Packages less than 10 lines of code are easily copied into your own project and may not warrant the additional supply chain risk of an external dependency. Found 1 instance in 1 package

Unpopular package QUALITY

This package is not very popular. Found 1 instance in 1 package

Version published 4 months ago

Maintainers 1

Weekly downloads 199

Readme



How can we mitigate

Registry Compromise

- **Package Signing**
 - Packages published to npm are signed with a public key
 - Rotation of the key & new signatures were created in August 2022 using the ECDSA algorithm alongside new npm CLI validation
- **SSRI, Caching & Lockfiles**
- **Integrity Checks**

How can we mitigate

Account Takeovers

- **Mandatory Login Verification**
Everyone - March 2022
- **Mandatory 2FA**
Top-100 Maintainers - February 2022
Top-500 Maintainers - May 2022
High Impact - November 2022
- **Improved 2FA Experience**
ex. Self-Serve Dashboard, Recovery Codes, Multiple Keys, Org-wide Management, WebAuthn
- **Improved Login Experience**
ex. npm login & npm publish (web login flow)
- **Investments in Support & Authentication**
ex. playbooks protecting against social engineering, identity verification & automation



High Impact Packages / Maintainers:
1 million+ weekly downloads or 500+ dependants

How can we mitigate **Malware**

- **Focus on package contents**
- **Active scanning for known patterns**
 - AI models tracking behaviours
- **Automated takedowns/advisories**
 - Partnerships with security experts & reporting APIs

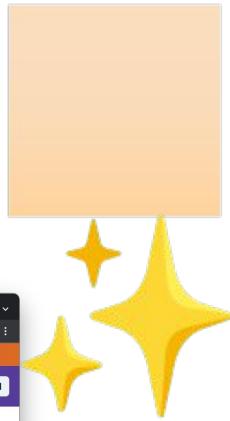
There are *other*

Threats

- **Noise**
- **Confusion**
- **Obfuscation**
- **Mutability**



Report malware



The screenshot shows the Socket dashboard interface. On the left, a sidebar menu includes: Overview, Alerts, Dependencies, Repositories, Reports (selected), Analytics, People, Security Policy, License Policy, Settings, Docs, Threat Feed, Changelog, and Get help. At the bottom of the sidebar is a button to 'Install Chrome Extension'. The main content area displays a report for npm package #72f76380-0a50-4a63-90e4-1330d59910bb, created on 2/27/2025 at 7:01:58 AM. The package has 35 dependencies, 4 maintainers, and 521 versions. The 'Reports' tab is selected, showing a list of alerts. One alert is highlighted as Critical: 'Critical CVE Lodash' with the note 'Contains a Critical Common Vulnerability and Exposure (CVE)'. Other alerts listed include 'High CVE Lodash', 'Socket optimized override available side-channel', 'Socket optimized override available safe-buffer', 'Socket optimized override available es-define-property', 'Socket optimized override available safer-buffer', and 'High CVE path-to-regexp'.

The screenshot shows the npm package overview for npm@4.0.2. The page features a purple header with the Socket logo and a 'Big news!' banner about Socket AI. Below the header, there's a search bar and navigation links for Product, Resources, Docs, Pricing, and npm. The main content area includes a summary card with metrics: Supply Chain Security (67), Quality (76), Maintenance (98), Vulnerabilities (60), and License (41). It also lists several security findings: 'CVE' (with 4 instances found), 'Bin script shell injection' (Supply Chain Risk, 1 instance found), 'Network access' (Supply Chain Risk, 8 instances found), 'Shell access' (Supply Chain Risk, 1 instance found), and 'Uses eval' (Supply Chain Risk, 1 instance found). At the bottom, there's a chart showing weekly downloads (4,700,416) with a 8.05% increase over the previous week.

Security - @darcy

Nondeterminism & Mutability

(ex. feature parity, remote third-party packages, install scripts etc.)

Example: Create React App Project

package.json post-initialization...

```
"dependencies": {  
  "@testing-library/jest-dom": "^5.16.5",  
  "@testing-library/react": "^13.4.0",  
  "@testing-library/user-event": "^13.5.0",  
  "react": "^18.2.0",  
  "react-dom": "^18.2.0",  
  "react-scripts": "5.0.1",  
  "web-vitals": "^2.1.4"  
}
```

 7 Direct Dependencies

Example:

Create React App Project

number of “dependencies” installed (no configuration)



yarn v1.22.19

1,256



pnpm v7.26.3

1,937



npm v9.4.2

1,408



bun v0.5.5

1,386

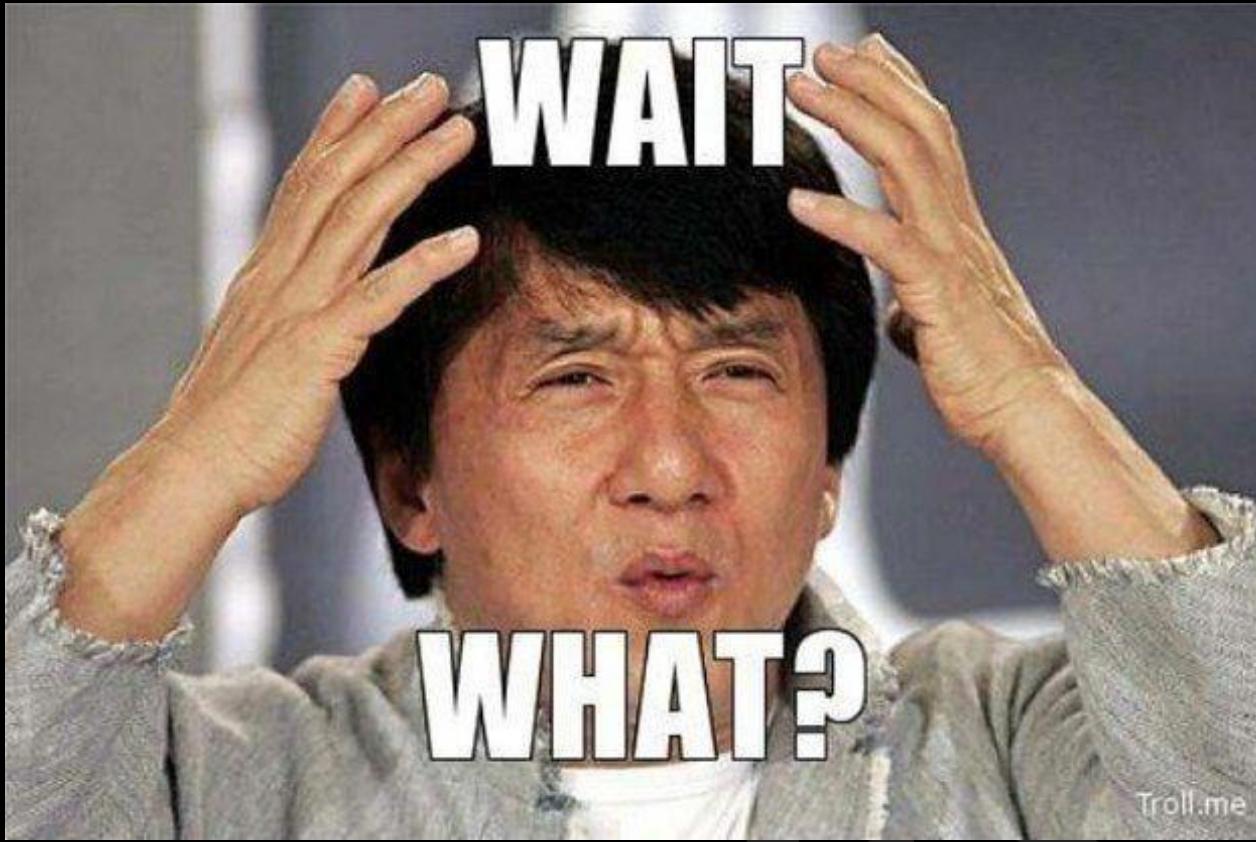


deno v1.3.1

1,083



⚠ There is a **-+ ~850** diff in dependencies!



Troll.me

Key:

Package managers treat these things (& more) differently...

Development Dependencies

Optional Dependencies (including environment-specific conditions)

Bundled Dependencies

Peer Dependencies

Overrides / Resolutions

Lifecycle Scripts

LIFE IS LIKE AN NPM INSTALL



YOU NEVER KNOW WHAT YOU'RE GONNA GET

imgflip.com

Avoid

Mutable Package References

- **Distribution Tags**
ex. "pkg@latest"
- **Remote Tarball URLs**
ex. "https://example.com/file.tgz"
ex. "https://example.com/"
- **Remote Git Repository URLs**
ex. "https://github.com/user/repo.git"
ex. "https://example.com/"



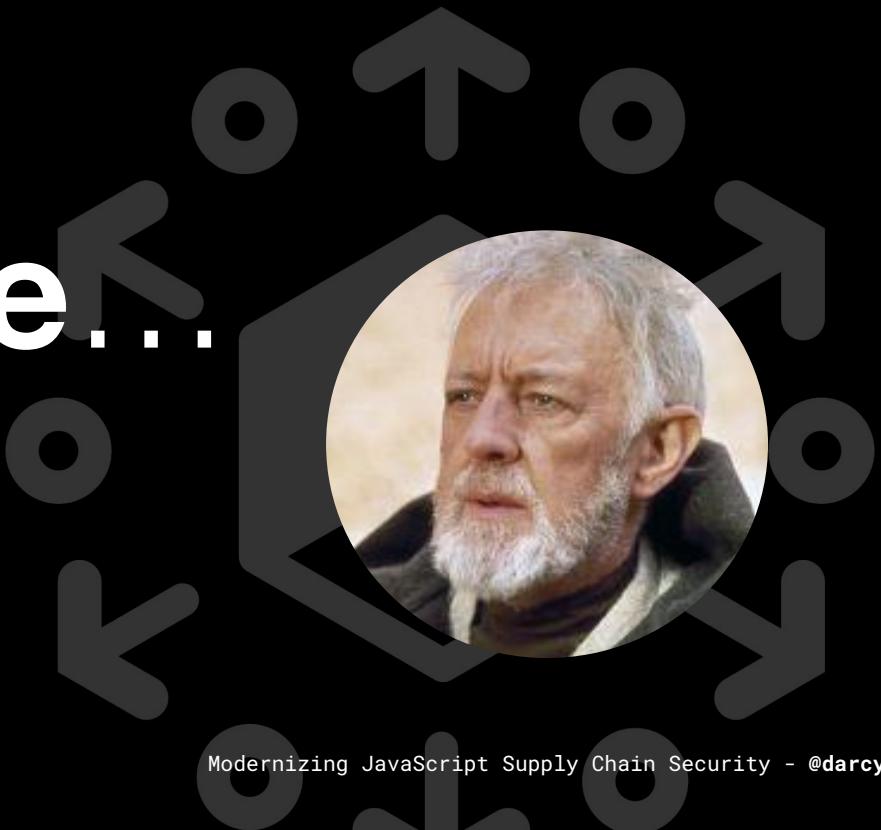
Package Documents reference data that is both **mutable** and **immutable*** & package metadata **is not validated**

Use

Lockfiles

- Contains:
 - **Integrity Values (SSRI)**
 - **Resolved References**
 - **Tree Shape**

There is **hope...**

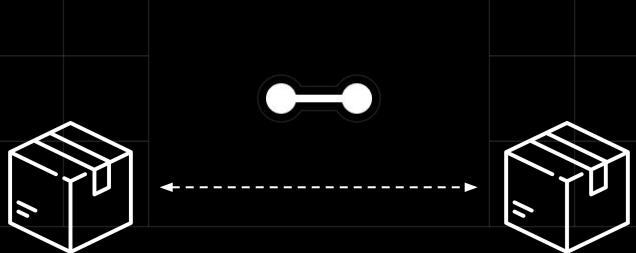


 vlt /vōlt/

vlt.sh

Introducing “reproduce”

```
$ npx reproduce <pkg>
```



<https://blog.vlt.sh/blog/reproduce>

The screenshot shows a dark-themed blog post. At the top right is a navigation bar with links for Product, Docs, Blog, and Company. The main title is "Is your package truly reproducible?" with a subtitle "vlt /volt/". Below the title is a large, blurry image of a person's face. The date "Tuesday, February 25, 2025" is at the bottom left of the image. The main content features a large, bold title: "Reproducibility vs. Provenance: Trusting the JavaScript Supply Chain". Below the title is a bio for "Drew Clarke" with a GitHub icon and the handle "@drewvlt". A section titled "Reproducibility vs. Provenance: Trusting the JavaScript Supply Chain" is described as follows: "The security and trustworthiness of the JavaScript package ecosystem have been under scrutiny for years. With growing concern over software supply chain attacks, the industry has turned down an avenue of tracking where packages come from, how they're built, and ensuring transparency. But provenance alone isn't enough. Enter `reproduce`, a new open-source tool designed to independently verify whether a published npm package can be faithfully rebuilt from its published source. Unlike provenance systems that merely associate a package with a build environment which can be somehow altered maliciously, `reproduce` goes a step further, empirically".

DSS

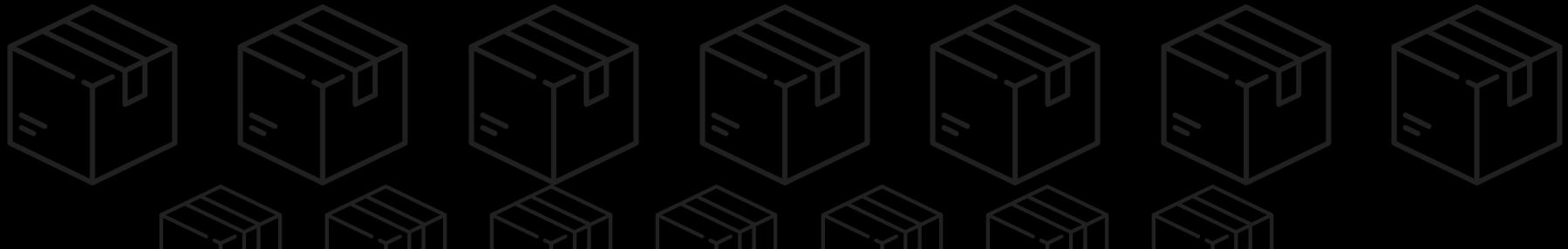
Dependency Selector Syntax



Dependency Selector Syntax (DSS)

<https://docs.vlt.sh/packages/query>

- CSS Selectors 4 Spec (syntax & operators)
- Answer complex, multi-faceted questions about dependencies, their relationships & associative metadata
- Consolidates redundant logic of similar, legacy list commands



Dependency Selector Syntax (DSS)

<https://docs.vlt.sh/packages/query>

```
*  
:root > *  
:root > .prod  
:root > .dev  
:root > * > .peer  
.workspace  
.workspace > .workspace  
.workspace:has(*.peer)  
[name=@vltpkg/query]  
#lodash  
#lodash[version~>1.2]  
*:empty  
*:has(*)  
*:not(:empty)  
*:type(git)
```

- all deps
- all direct deps
- direct production deps
- direct development deps
- any peer dep of a direct deps
- any workspace dep
- all workspaces that depend on another workspace
- all workspaces that have peer deps
- deps named "@vltpkg/query"
- dep named "lodash"
- deps named "lodash" & version starting w/ "1.2"
- deps with no other deps (ie. "leaf" nodes)
- has any deps
- "" - equivalent to the above
- querying for all git dependencies

Dependency Selector Syntax (DSS)

<https://docs.vlt.sh/packages/query>

```
// find all dependencies with specific licenses
*[license="MIT"], *[license="ISC"]

// find all dependencies that have a node.engines property set
*:attr(engines, [node])

// find all dependencies that have defined react as an optional peer
*:has(#react):not(:attr(peerDependenciesMeta, react, [optional]))

// find all dependencies that have myself as a contributor
*:attr(contributors, [email=luke@lukekarrys.com])

// find all references to "install" scripts
*[scripts=install],
*[scripts=postinstall],
*[scripts=preinstall]
```

Dependency Selector Syntax (DSS)

<https://docs.vlt.sh/packages/query>

:semver(<spec>, <function>, <selector>) - semver comparator to [version]

- **spec** - a semver version or range
- **selector** - an attribute selector for each node (defaults to [version])
- **functions** - a semver method to apply, one of: `satisfies`, `intersects`, `subset`, `gt`, `gte`, `gtr`, `lt`, `lte`, `ltr`, `eq`, `neq` or the special function `infer` (default `infer`)

:outdated - have newer versions available

:outdated(<type>) - have a specific type of version available
ex. "MAJOR", "MINOR", "PATCH", defaults to "ANY"

Command Line

```
$ vlt query "<selector>"
```

```
ex. vlt query "#react" | jq 'map(.name + "@" + .version)'
```

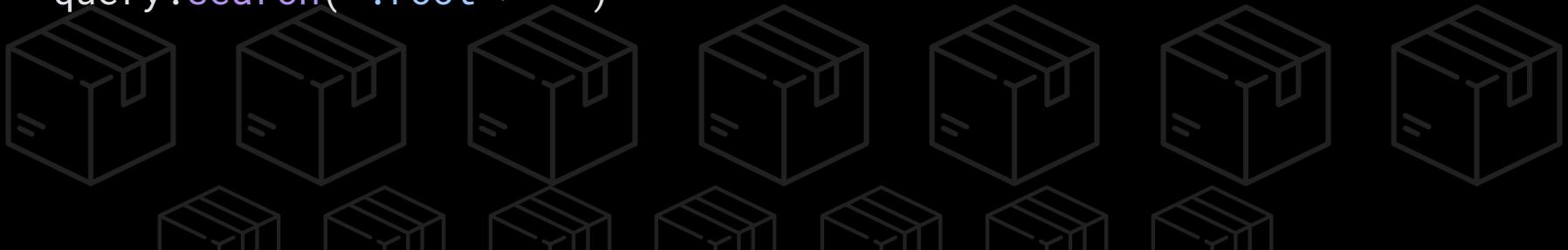


Programmatic Usage

`@vltpkg/query`

```
import { actual } from '@vltpkg/graph'
import { Query } from '@vltpkg/query'

const graph = await actual.load({ projectRoot: process.cwd() })
const query = new Query([...graph.nodes.values()])
query.search(':root > *')
```



GUI

The screenshot shows the vlt Explorer application running in a macOS-style window. The main area is titled "Explore" and displays information about the "postcss" package version 8.4.49. The interface includes a sidebar with a dashboard, queries, and project lists, and a central panel for viewing package details and dependencies.

Explore View Details:

- Selected Item:** postcss v8.4.49 (published by ai, 6,729,473 weekly downloads)
- Dependents:** postcss-import v15.1.0, postcss-js v4.0.1, postcss-load-config v4.0.2, postcss-nested v6.2.0, tailwindcss v3.4.15
- Dependencies:** nanoid v3.3.7, picocolors v1.1.1, source-map-js v1.2.1
- Manifest:** Contains the package's JSON manifest file.
- Versions:** Shows available versions of the package.

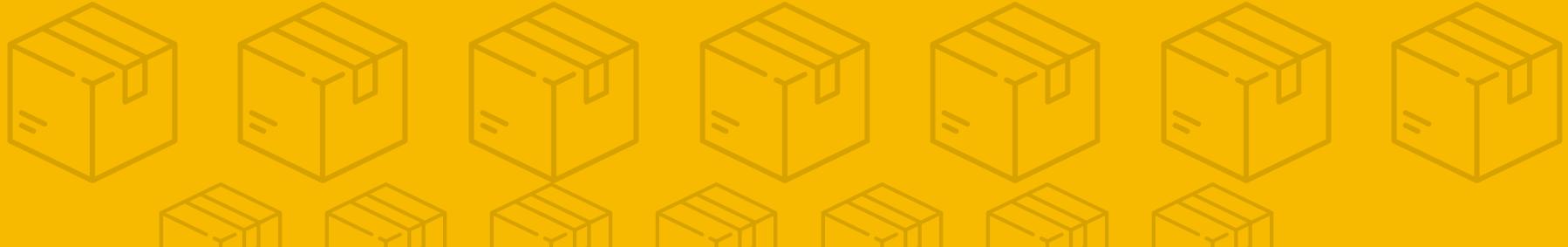
Left Sidebar (Dashboard):

- Dashboard
- Queries (2)
- Projects
 - @vtpkg/registry (June 28th, 2024 | 06:27)
 - apollo-client-devtools (June 17th, 2024 | 10:41)
 - d3-pprof (December 11th, 2024 | 03:07)
 - gsap (August 19th, 2024 | 04:55)
 - npm (December 10th, 2024 | 04:47)

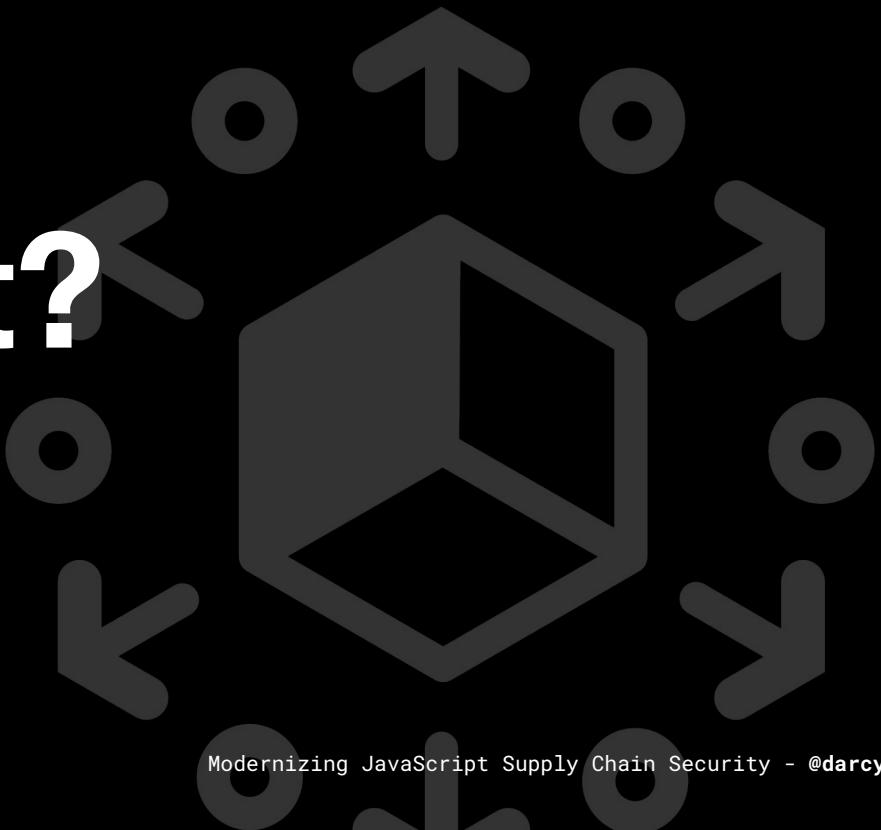
Bottom Navigation:

- Documentation

Demo



What's Next?



Security-Focused Pseudo States

```
:suspicious (ref. https://socket.dev/alerts/suspiciousStarActivity)
:undesirable (ref. https://socket.dev/alerts/troll)
:unstable (ref. https://socket.dev/alerts/unstableOwnership)
:abandoned (ref. https://socket.dev/alerts/missingAuthor)
:trivial (ref. https://socket.dev/alerts/trivialPackage)
:unknown (ref. https://socket.dev/alerts/newAuthor)
:unpopular (ref. https://socket.dev/alerts/unpopularPackage)
:deprecated (ref. https://socket.dev/alerts/deprecated)
:unmaintained (ref. https://socket.dev/alerts/unmaintained)
:shrinkwrap (ref. https://socket.dev/alerts/shrinkwrap)
:obfuscated (ref. https://socket.dev/alerts/obfuscatedFile)
:tracker (ref. https://socket.dev/alerts/telemetry)
:eval (ref. https://socket.dev/alerts/usesEval)
```



```
:scripts (ref. https://socket.dev/alerts/installScripts)
:shell (ref. https://socket.dev/alerts/shellAccess)
:native (ref. https://socket.dev/alerts/hasNativeCode)
:confused (ref. https://socket.dev/alerts/manifestConfusion)
:network (ref. https://socket.dev/alerts/networkAccess)
:debug (ref. https://socket.dev/alerts/debugAccess)
:dynamic (ref. https://socket.dev/alerts/dynamicRequire)
:fs (ref. https://socket.dev/alerts/filesystemAccess)
:entropic (ref. https://socket.dev/alerts/highEntropyStrings)
:env (ref. https://socket.dev/alerts/envVars)
:minified (ref. https://socket.dev/alerts/minifiedFile)
:deprecated (ref. https://socket.dev/alerts/deprecated)
```

Policies

package.json

```
"audit": {  
  "policies": [  
    {  
      "name": "Vulnerable",  
      "type": "error",  
      "query": ":vulnerable"  
    },  
    {  
      "name": "Peer Conflicts",  
      "type": "error",  
      "query": ".peer:not(:deduped)"  
    },  
    {  
      "name": "Deprecated",  
      "type": "warn",  
      "query": ":deprecated"  
    },  
    ...  
  ]  
}
```

Shape:

```
{  
  "name": "<name>",  
  "type": "<log|warn|error>",  
  "query": "<selector>"  
}
```



Key Takeaways:

- **New tools** are here & more are coming soon...
- **Feature-rich** tools > fast tools
- Every new piece of **metadata** makes the **query language** & dep graph traversal more **powerful**
- **Accurate** information in supply chains is mission critical

Thank you!

Bluesky:

@darcyclarke.me

Twitter / X:

@darcy

GitHub:

@darcyclarke

Website:

darcyclarke.me

Bluesky:

@vltpkg.sh

Twitter / X:

@vltpkg

GitHub:

@vltpkg

Website:

vl.sh



Please give feedback!