IPC - Berlin - May 22, 2023

Get A Grip On Your Project's Supply Chain

Nils Adermann
@naderman

Private Packagist
https://packagist.com

Supply Chain?

134
B21
KRUPP

18
MSC HOME TERMINAL
17
MSC HOME TERMINAL
16
MSC HOME TERMINAL
15
14
13
MSC TRIESTE
OOCL BELGIUM

67
65
63
61
SDI
OVERHEAD DOOR
074

https://commons.wikimedia.org/wiki/File:Geely_assembly_line_in_Beilun,_Ningbo.JPG

Supply Chain

A supply chain is a complex logistics system that consists of facilities that convert raw materials into finished products which are later distributed to end consumers or end customers.

https://en.wikipedia.org/wiki/Supply_chain

Supply Chain - But for Software?!

Raw materials	Source code
Refining, processing, constructing	Build process
Product components	Dependencies, Hardware, Network
Assembly, logisitics	Package management
Quality assurance	QA / CI Service
Order fullfillment	Deployment process

Take with a grain of salt - this comparison will only take you so far

Software Supply Chain

A software supply chain is composed of the components, libraries, tools, and processes used to develop, build, and publish a software artifact.

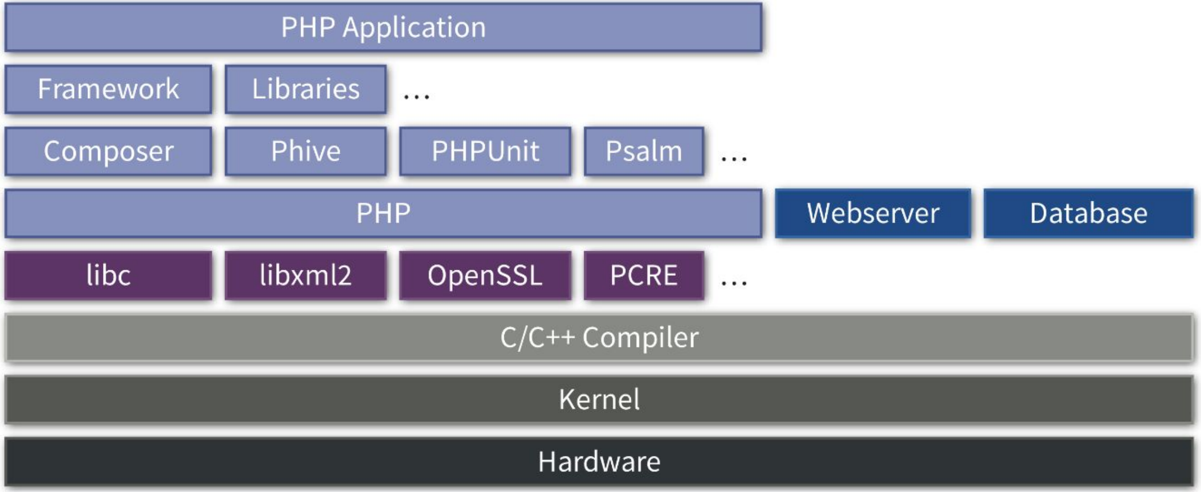
https://en.wikipedia.org/wiki/Software_supply_chain

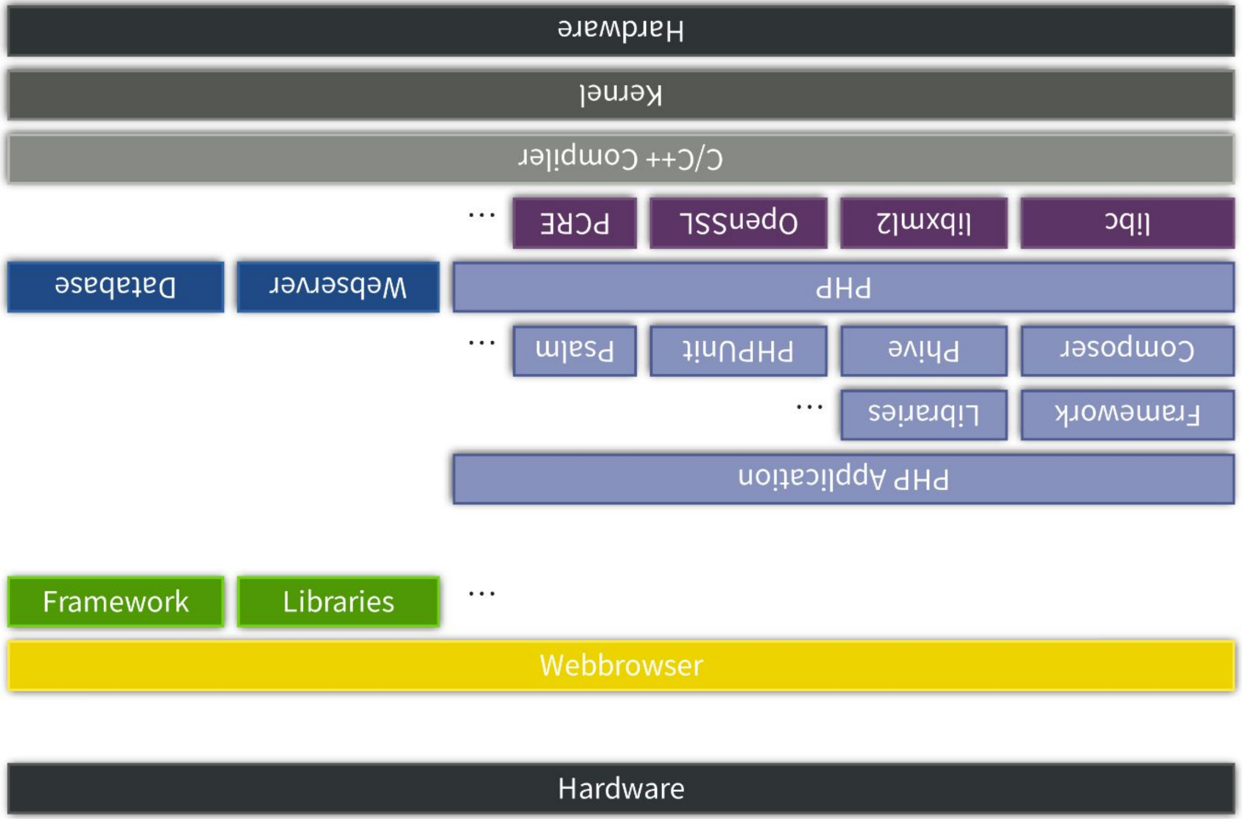
Software Supply Chain

In other words:

The “full-stack” and all processes & tools involved in making and assembling it

Full-stack


PHP Application
Framework
Libraries
…
Composer
Phive
PHPUnit
Psalm
…
PHP
Webserver
Database
libc
libxml2
OpenSSL
PCRE
…
C/C++ Compiler
Kernel
Hardware



PHP Application

Framework | Libraries | …

Composer | Phive | PHPUnit | Psalm | …

PHP | Webserver | Database

libc | libxml2 | OpenSSL | PCRE | …

C/C++ Compiler

Kernel

Hardware

Framework | Libraries | …

Webbrowser

Hardware

Why should you care?

- Business Continuity
  - What if your datacenter is on fire?
  - What if your CI platform goes out of business?
  - What if a dependency isn’t maintained anymore?
  - What if a dependency is deleted?
- Security
  - Supply Chain Attacks

Supply Chain Attacks

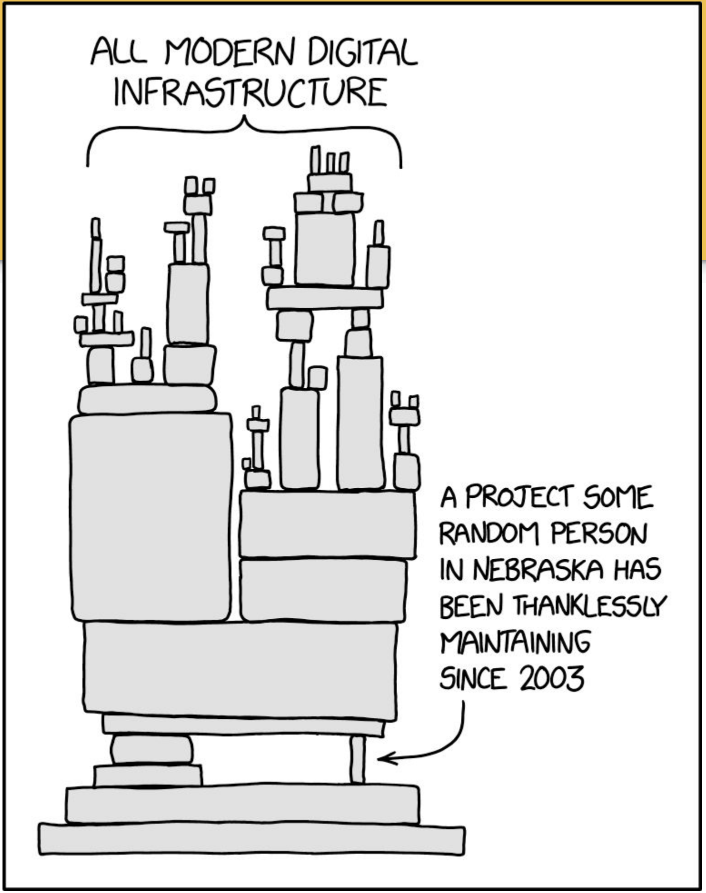
- Heartbleed - https://heartbleed.com/
  - The Heartbleed bug allows anyone on the Internet to read the memory of the systems protected by the vulnerable versions of the OpenSSL software. This compromises the secret keys used to identify the service providers and to encrypt the traffic, the names and passwords of the users and the actual content. This allows attackers to eavesdrop on communications, steal data directly from the services and users and to impersonate services and users.

Supply Chain Attacks

- $2,000 donations per year to OpenSSL
- $841 in 3 days after Heartbleed

- Creation of the Core Infrastructure Initiative at the Linux Foundation, now Open Software Security Foundation (OpenSSF)
  - https://openssf.org/
  - > $10 million raised by 2021

- Germany: Sovereign Tech Fund
  - https://sovereigntechfund.de

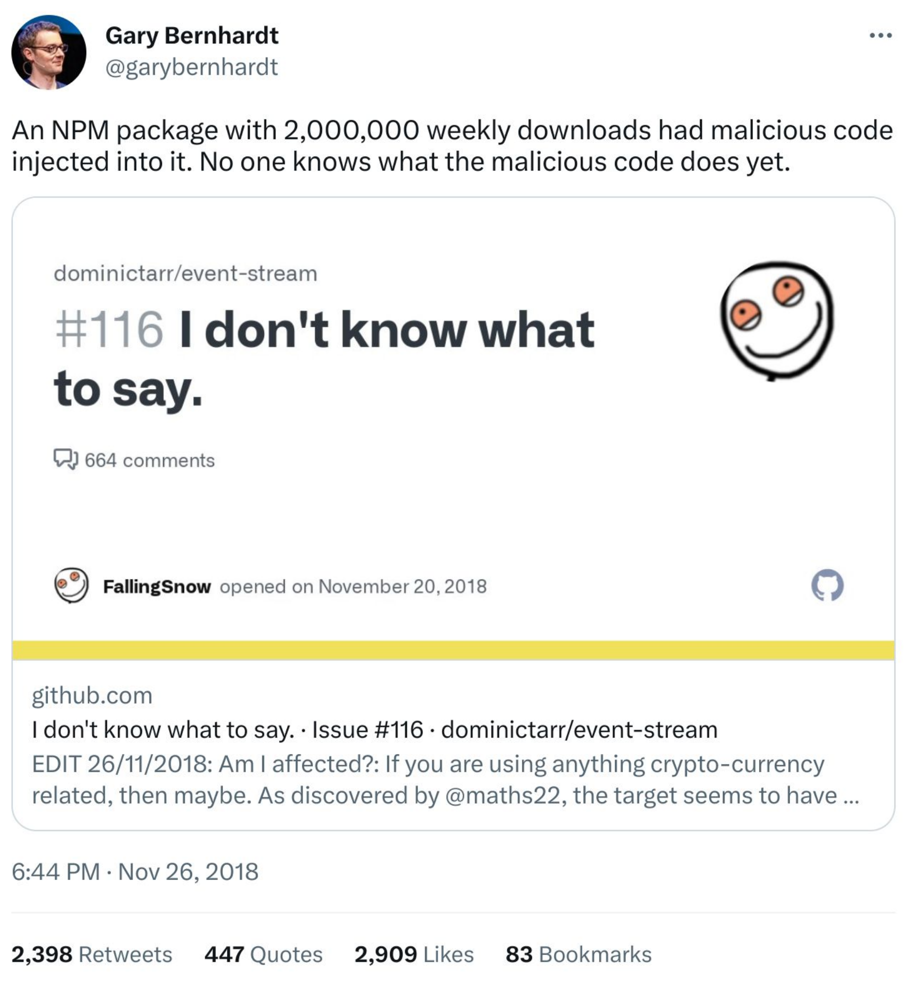

ALL MODERN DIGITAL INFRASTRUCTURE
A PROJECT SOME RANDOM PERSON IN NEBRASKA HAS BEEN THANKLESSLY MAINTAINING SINCE 2003

PRIVATE PACKAGIST

Supply Chain Attacks

- Stuxnet
  - uncovered in 2010, likely as old as 2005
  - combination of 4 zero-days, Windows, Siemens Step7, introduced on USB drives
  - targetted PLCs (programmable logic controllers) with a rootkit
  - likely to have been built by USA and Israel to damage Iranian nuclear program
- SolarWinds Orion / 2020 United States federal government data breach
  - attackers gained entry to a build system, likely through a compromised Office 365 account
  - modified software updates to include remote access on any machine installing Orion
  - 18,000 customers including many parts of the US government affected
  - likely Russian attackers
  - discovered in December ‘20 after breach Sep ‘19

Ownership of a dependency was transferred to a bad actor



Gary Bernhardt
@garybernhardt

An NPM package with 2,000,000 weekly downloads had malicious code injected into it. No one knows what the malicious code does yet.

dominictarr/event-stream

#116 **I don't know what to say.**

664 comments

FallingSnow opened on November 20, 2018

github.com

I don't know what to say. · Issue #116 · dominictarr/event-stream

EDIT 26/11/2018: Am I affected?: If you are using anything crypto-currency related, then maybe. As discovered by @maths22, the target seems to have ...

6:44 PM · Nov 26, 2018

2,398 Retweets **447** Quotes **2,909** Likes **83** Bookmarks

https://twitter.com/garybernhardt/status/1067111872225136640

Supply Chain Attacks

- Depublication of left-pad
    - https://qz.com/646467/how-one-programmer-broke-the-internet-by-deleting-a-tiny-piece-of-code
- PyPi Typosquatting with malicious code
    - https://blog.phylum.io/phylum-discovers-revived-crypto-wallet-address-replacement-attack/
- Public Travis CI Logs (Still) Expose Users to Cyber Attacks
    - https://blog.aquasec.com/travis-ci-security
- Malicious commits made to php-src in the name of Rasmus Lerdorf and Nikita Popov
    - https://news-web.php.net/php.internals/113838

Other Supply Chain Problems

- Jira: Atlassian customers frustrated by weeks-long outage, lack of communication from company
  - https://www.techrepublic.com/article/atlassian-customers-frustrated-by-weeks-long-outage-lack-of-communication-from-company/
- Following theft of GitHub OAuth tokens from Heroku, GitHub resets tokens but Salesforce takes weeks to reset passwords and restore functionality
  - https://www.zdnet.com/article/heroku-to-begin-user-password-reset-almost-a-month-after-github-oauth-token-theft/

Supply Chain Attacks

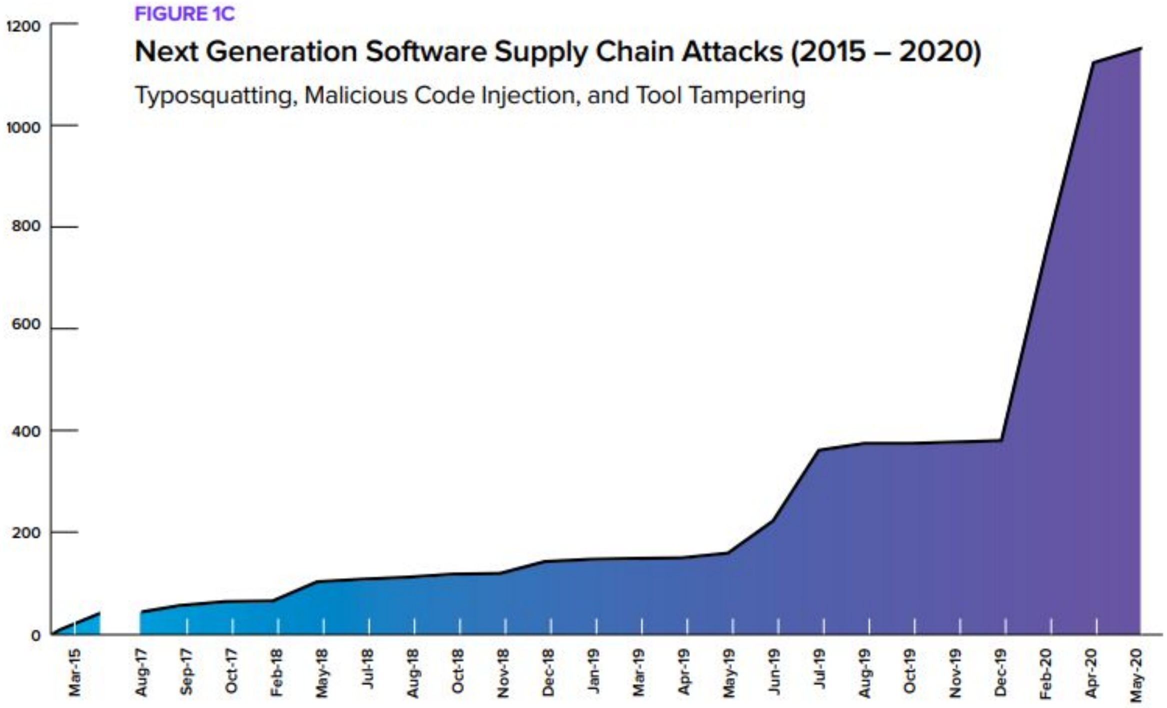


FIGURE 1C

Next Generation Software Supply Chain Attacks (2015 – 2020)

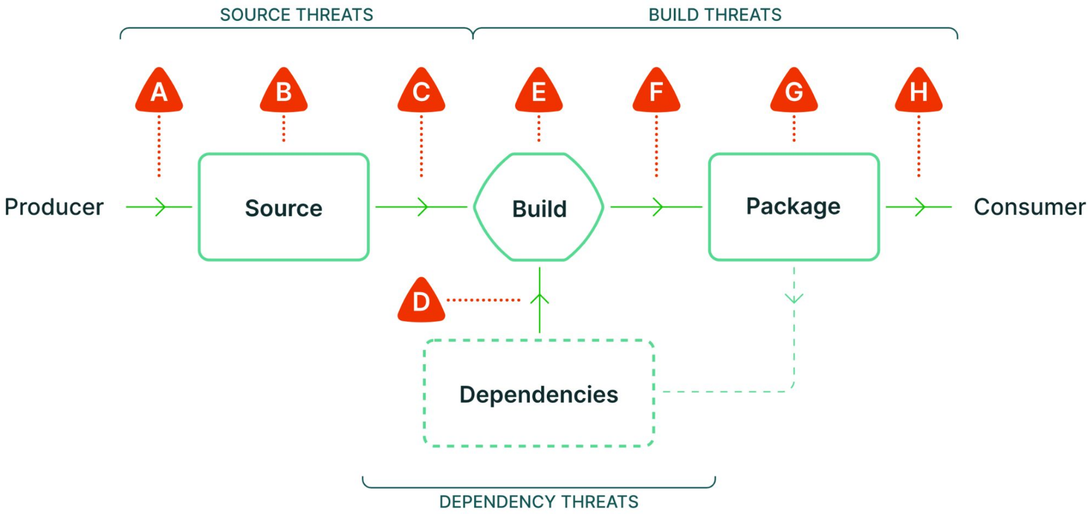
Typosquatting, Malicious Code Injection, and Tool Tampering

“2020 State of the Software Supply Chain” by sonatype

https://www.sonatype.com/hubfs/Corporate/Software%20Supply%20Chain/2020/SON_SSSC-Report-2020_final_aug11.pdf#page=7

Supply Chain Attacks

2021 Google Introduces SLSA “Supply-chain Levels for Software Artifacts” - https://slsa.dev/



SOURCE THREATS

A Submit unauthorized change

B Compromise source repo

C Build from modified source

DEPENDENCY THREATS

D Use compromised dependency

BUILD THREATS

E Compromise build process

F Upload modified package

G Compromise package repo

H Use compromised package

PRIVATE PACKAGIST

May 12, 2021
US Government acts: Executive Order 14028

- Introduces requirement for SBOM (Software Bill of Materials)
- Linux Foundation SPDX SBOMs
  - https://spdx.dev/
  - Can be exported directly from GitHub dependency graph
- OWASP CycloneDX
  - https://cyclonedx.org/
  - Composer plugin: `cyclonedx/cyclonedx-php-composer`

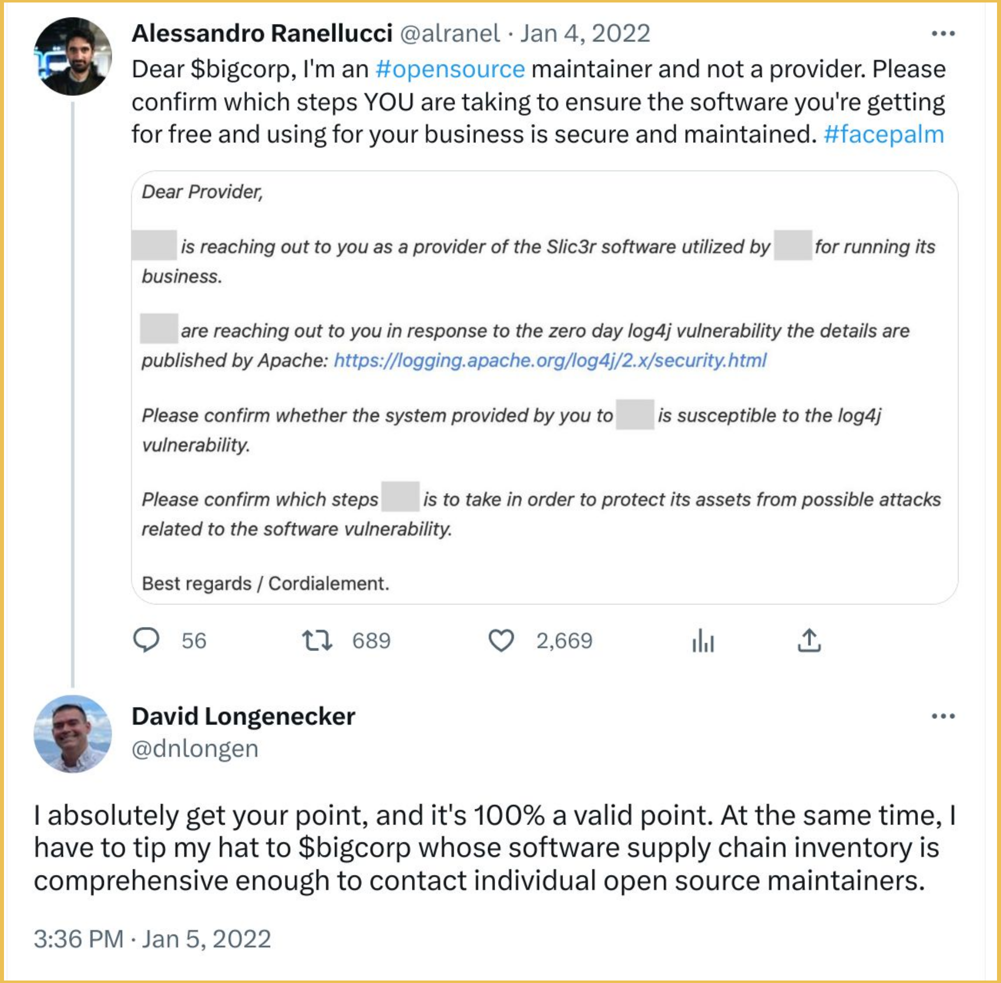
Get a Grip On Your Project’s Supply Chain

- Identifying your supply chain and documenting it
  - all tools and dependencies used: SBOMs
  - all services used: Who are the vendors? Use checklists to collect information
  - all processes and infrastructure used

- Risk analysis
  - probability of failure
  - impact of failure

Supply Chain Attacks

- Log4Shell
  - https://en.wikipedia.org/wiki/Log4Shell
  - Log4j vulnerability, standard Java logging library
  - existed 2013 - November 24, 2021
  - Arbitrary code execution, extremely widely used, CVSS Score 10/10



Alessandro Ranellucci @alranel · Jan 4, 2022

Dear $bigcorp, I'm an #opensource maintainer and not a provider. Please confirm which steps YOU are taking to ensure the software you're getting for free and using for your business is secure and maintained. #facepalm

> *Dear Provider,*
>
> [redacted] *is reaching out to you as a provider of the Slic3r software utilized by* [redacted] *for running its business.*
>
> [redacted] *are reaching out to you in response to the zero day log4j vulnerability the details are published by Apache: https://logging.apache.org/log4j/2.x/security.html*
>
> *Please confirm whether the system provided by you to* [redacted] *is susceptible to the log4j vulnerability.*
>
> *Please confirm which steps* [redacted] *is to take in order to protect its assets from possible attacks related to the software vulnerability.*
>
> Best regards / Cordialement.

David Longenecker
@dnlongen

I absolutely get your point, and it's 100% a valid point. At the same time, I have to tip my hat to $bigcorp whose software supply chain inventory is comprehensive enough to contact individual open source maintainers.

3:36 PM · Jan 5, 2022

https://twitter.com/dnlongen/status/1478737214179844100

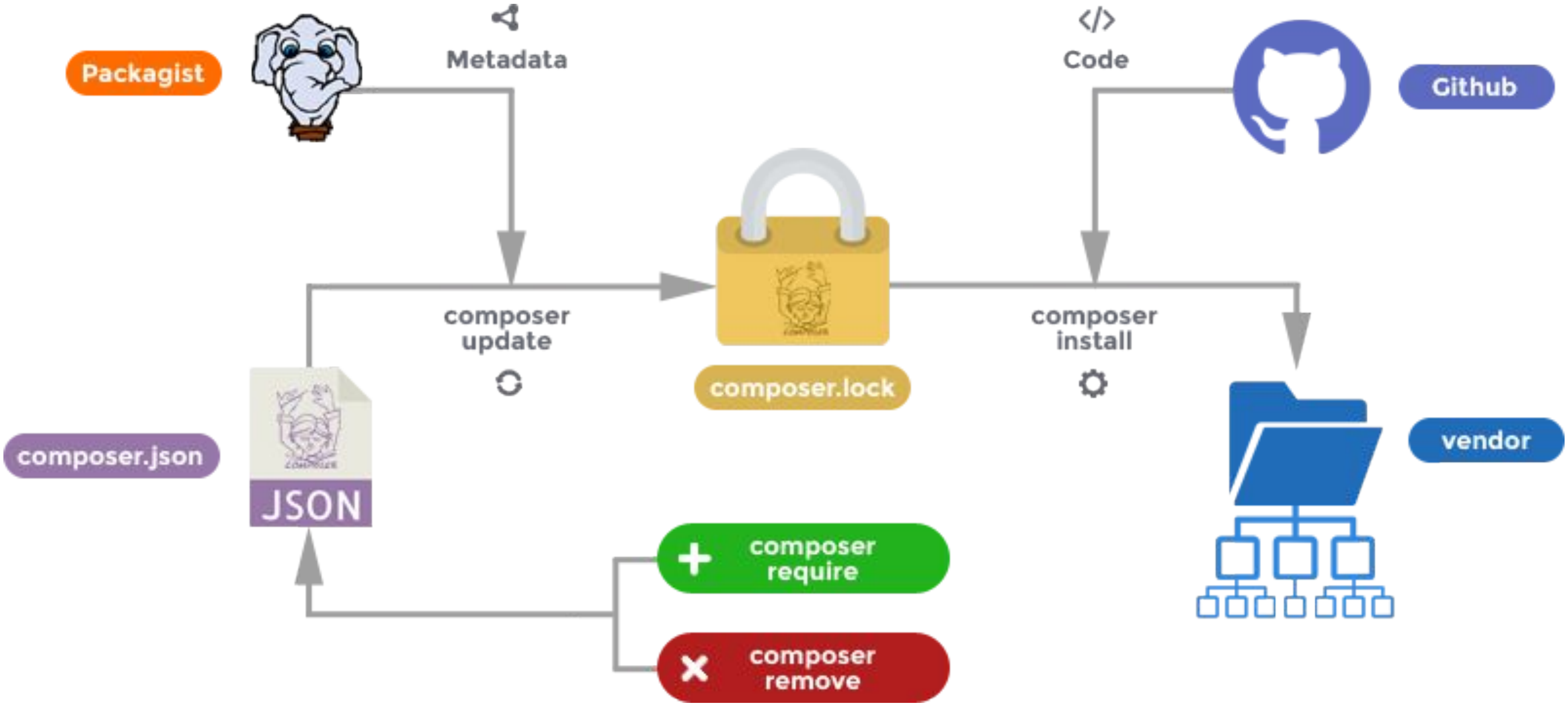
Get a Grip On Your Project’s Supply Chain

- Risk mitigation
  - Regularly identify and upgrade outdated software
    - automate as much as possible
  - Audit your vendors
    - You can’t do everything yourself and are likely going to be worse at e.g. following hardware security updates than a large cloud hoster
  - Select processes that reduce risk
    - deploy tested artifacts, rather than building during deploy which may differ from CI
    - prefer declarative state over modifying state over time

Composer Guide to Supply Chain Security

PRIVATE PACKAGIST

composer update vs. composer install



Packagist.org

- Metadata only
- No checksums for GitHub stored packages
  - https://github.com/sansecio/composer-integrity-plugin
- No signatures
  - https://www.drupal.org/project/infrastructure/issues/3325040 - Automatic Updates / TUF

- No way to upload code
- Packagist.org maintainer account takeover
  https://blog.packagist.com/packagist-org-maintainer-account-takeover/
  - Editing of source URLs no longer allowed beyond 50k installs

Supply Chain Attacks

- Apr 13, 2022: Composer Command Injection Vulnerability
  - https://blog.packagist.com/cve-2022-24828-composer-command-injection-vulnerability/
  - Code execution through Git or Mercurial branch names
- Apr 27, 2021: Composer Command Injection Vulnerability
  - https://blog.packagist.com/composer-command-injection-vulnerability/
  - Code execution through Mercurial repository URL injection
- Mar 11, 2021: Git Clone Security Vulnerability
  - https://blog.packagist.com/git-clone-security-vulnerability/
  - Git vulnerability on case insensitive filesystems can be exploited through Composer if you clone dependencies

So commit your vendor directory?

- Who here knows how to commit changes to the files?

So commit your vendor directory?

- Who here knows how to commit changes to the files?
  - `git add vendor/` will not delete files, can lead to bugs and security issues
  - Must use `git add -A vendor/`

- vendor directory contents can diverge from expected content
  - How do you verify vendor directory contents match the lock file?
    - e.g. are deleted packages really deleted?

- Managing conflicts in larger teams gets even harder than managing lock file contents

So commit your vendor directory?

- Bad Actor scenarios, e.g. disgruntled employee
    - Scenarios
        - Could place code in unmanaged directory in vendor looking like a dependency
        - Could modify code of existing package in vendor/

    - Would your review process catch these as part of a large update commit?
    - If not, do you have tooling to notice the discrepancy?
        - Is building this tooling less work/cheaper than using a private Composer repository?

Generally: **No, don't commit the vendor directory**

Use your own Composer repository

- Satis
- JFrog Artifactory
- Sonatype Nexus Repository
- Cloudsmith
- GitLab Package Registry
- …

- **Private Packagist**

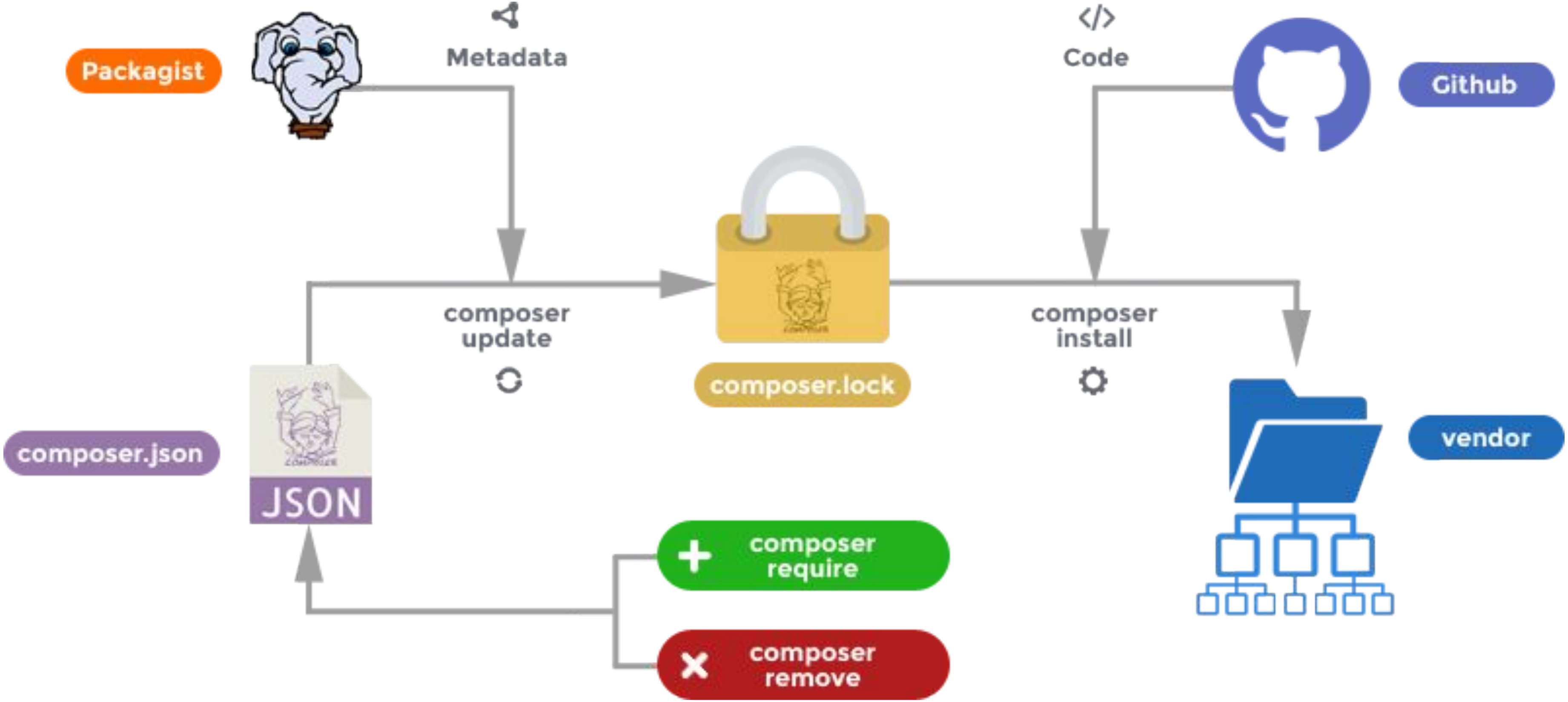
Private Packagist

- Stores a copy of all used versions of your dependencies
  - Safe from deletion
  - Safe from modification

- Serves package metadata and code

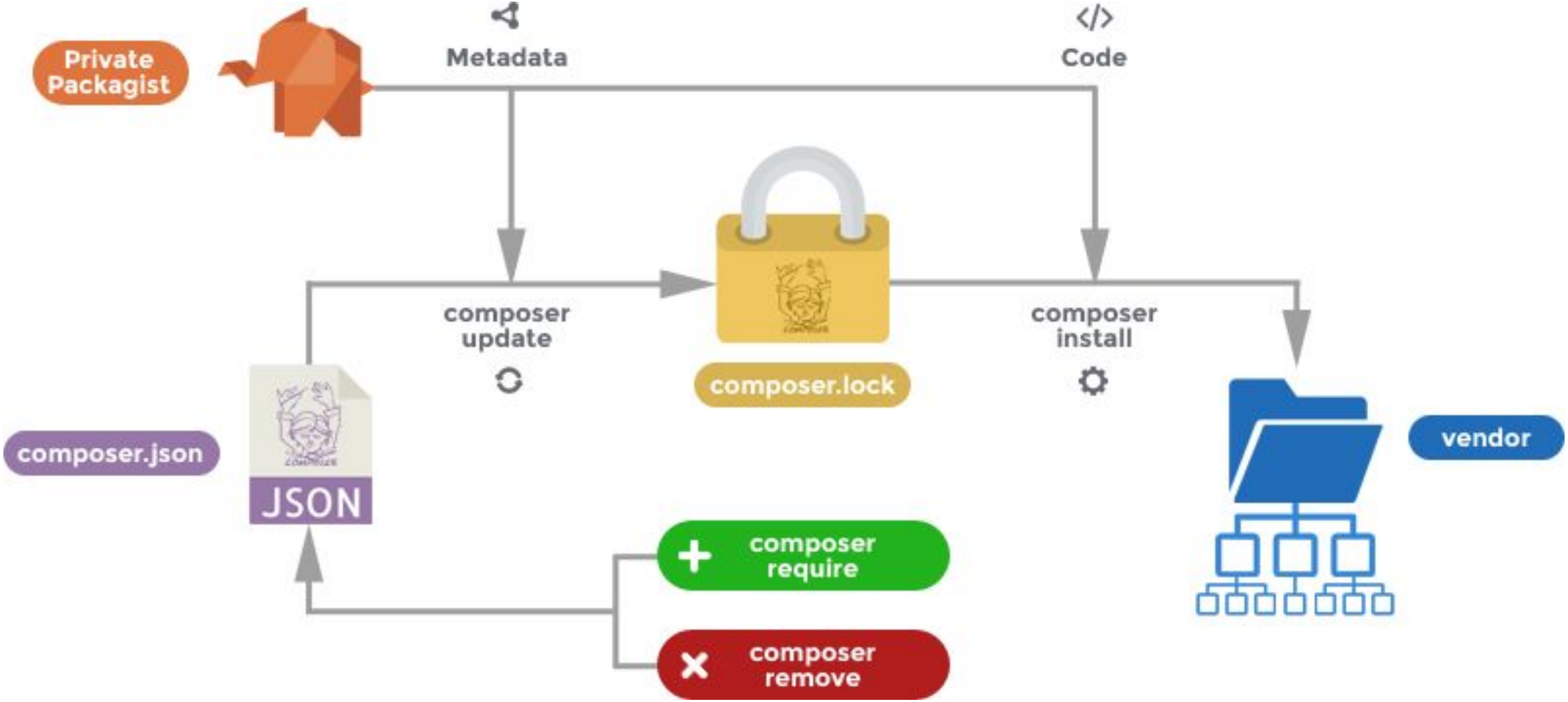
- Possible with some alternatives but usually with more effort and less convenience
  - e.g. copy all dependencies into git repositories, how do you keep those updated then?

Public packagist.org / GitHub



PRIVATE PACKAGIST

Private Packagist



Update Dependencies Frequently

- Set up a schedule or regular reminder to run dependency updates
- Set up alerting when vulnerabilities are discovered in your dependencies
  - GitHub Dependabot https://docs.github.com/en/code-security/dependabot/dependabot-alerts/about-dependabot-alerts
  - Snyk https://snyk.io/product/open-source-security-management/
  - **Private Packagist** Security Monitoring https://packagist.com/features/security-monitoring

Update Dependencies Frequently

Better yet: Automate your updates

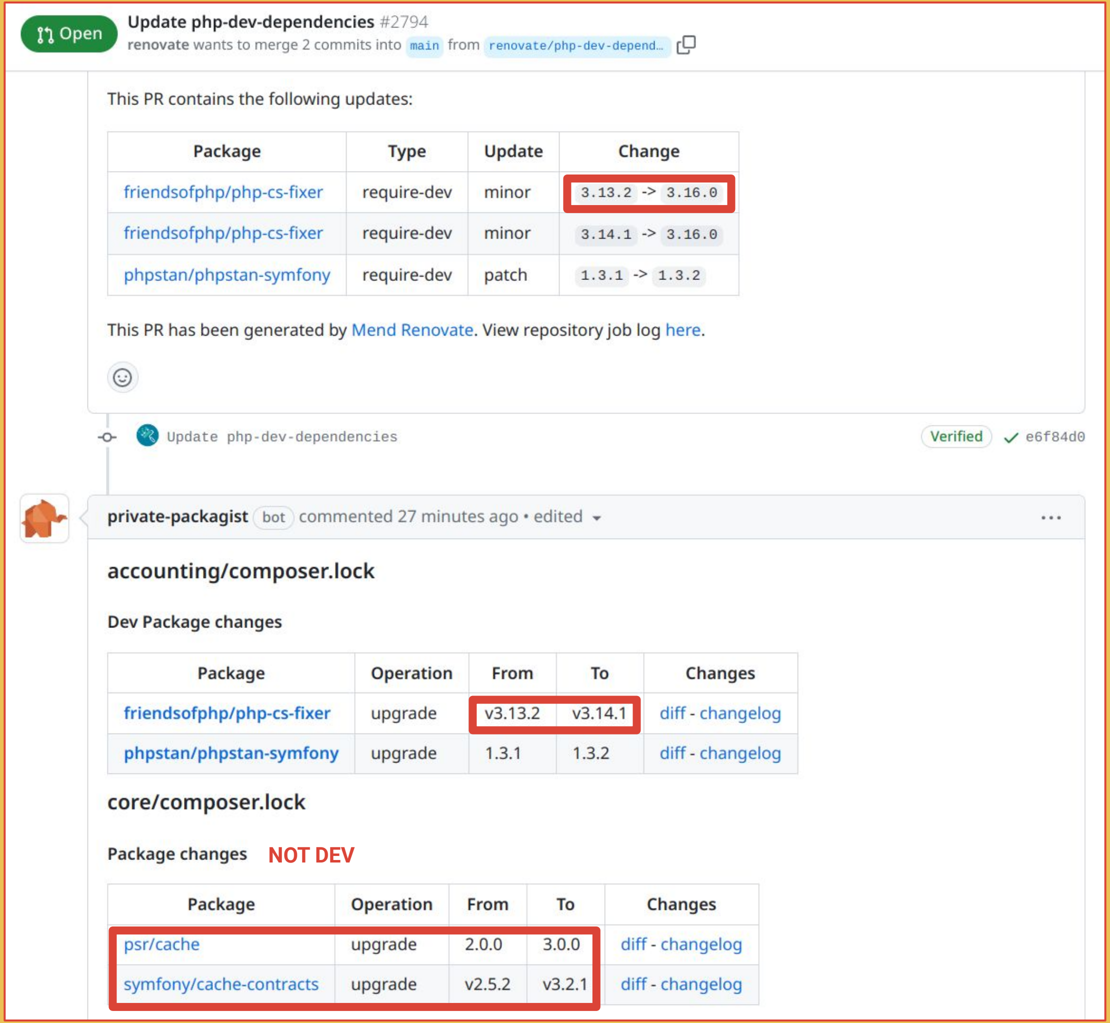
- Mend Renovate https://www.mend.io/renovate/
- GitHub Dependabot https://github.com/dependabot
- *(WIP: Private Packagist Automated Updates)*

Get a pull request anytime an update is necessary

Caution!

Private Packagist Update Review

GitHub
BitBucket
GitLab



Update php-dev-dependencies #2794

renovate wants to merge 2 commits into `main` from `renovate/php-dev-depend…`

This PR contains the following updates:

Package	Type	Update	Change
friendsofphp/php-cs-fixer	require-dev	minor	`3.13.2` -> `3.16.0`
friendsofphp/php-cs-fixer	require-dev	minor	`3.14.1` -> `3.16.0`
phpstan/phpstan-symfony	require-dev	patch	`1.3.1` -> `1.3.2`

This PR has been generated by Mend Renovate. View repository job log here.

`Update php-dev-dependencies` Verified ✓ `e6f84d0`

private-packagist bot commented 27 minutes ago • edited

accounting/composer.lock

Dev Package changes

Package	Operation	From	To	Changes
friendsofphp/php-cs-fixer	upgrade	v3.13.2	v3.14.1	diff - changelog
phpstan/phpstan-symfony	upgrade	1.3.1	1.3.2	diff - changelog

core/composer.lock

Package changes NOT DEV

Package	Operation	From	To	Changes
psr/cache	upgrade	2.0.0	3.0.0	diff - changelog
symfony/cache-contracts	upgrade	v2.5.2	v3.2.1	diff - changelog

Composer Plugins & Scripts

- Composer 2.2 introduced a requirement to explicitly enable plugins
  - `config.allow-plugins`
  - protects you from unintentionally executing malicious code before reviewing composer.lock changes

- Scripts & plugin selection is limited to root composer.json
  - Protects from attacks by malicious maintainers, dependency confusion or other accidental dependencies
  - You still have to review your lock file changes!

Recommended use of Composer in your Deployment Process

- commit composer.lock
- CI/CD
  - run composer install (not update!)
  - generate any potentially generated code
  - package everything into an archive
- deployment
  - upload to production servers, move in place
  - run composer check-platform-reqs
  - dump an optimized autoloader
  - switch webserver to use new code

Result

- no surprises in production
  - same dependency versions as tested
  - no risk of composer conflicts during deploy
  - code doesn’t change at runtime
- deploying to multiple servers
  - exact same state everywhere
  - no unnecessarily repeated work

Questions / Feedback?

Private Packagist
https://packagist.com

E-Mail: contact@packagist.com
Twitter: @naderman