PHP[TEK] 2024

Composer Guide to Supply Chain Security

Nils Adermann
@naderman

Private Packagist
https://packagist.com

Supply Chain?

PRIVATE PACKAGIST

MSC HOME TERMINAL
MSC HOME TERMINAL
MSC HOME TERMINAL
MSC TRIESTE
OOCL BELGIUM

67
65
63
61
SDI
074

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

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 / Tests / 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.

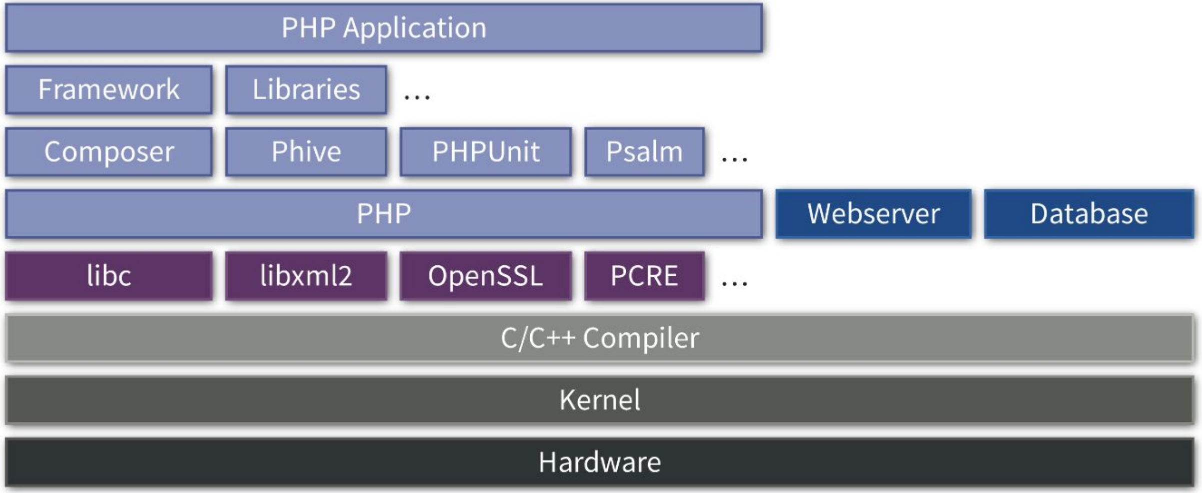
https://en.wikipedia.org/wiki/Software_supply_chain

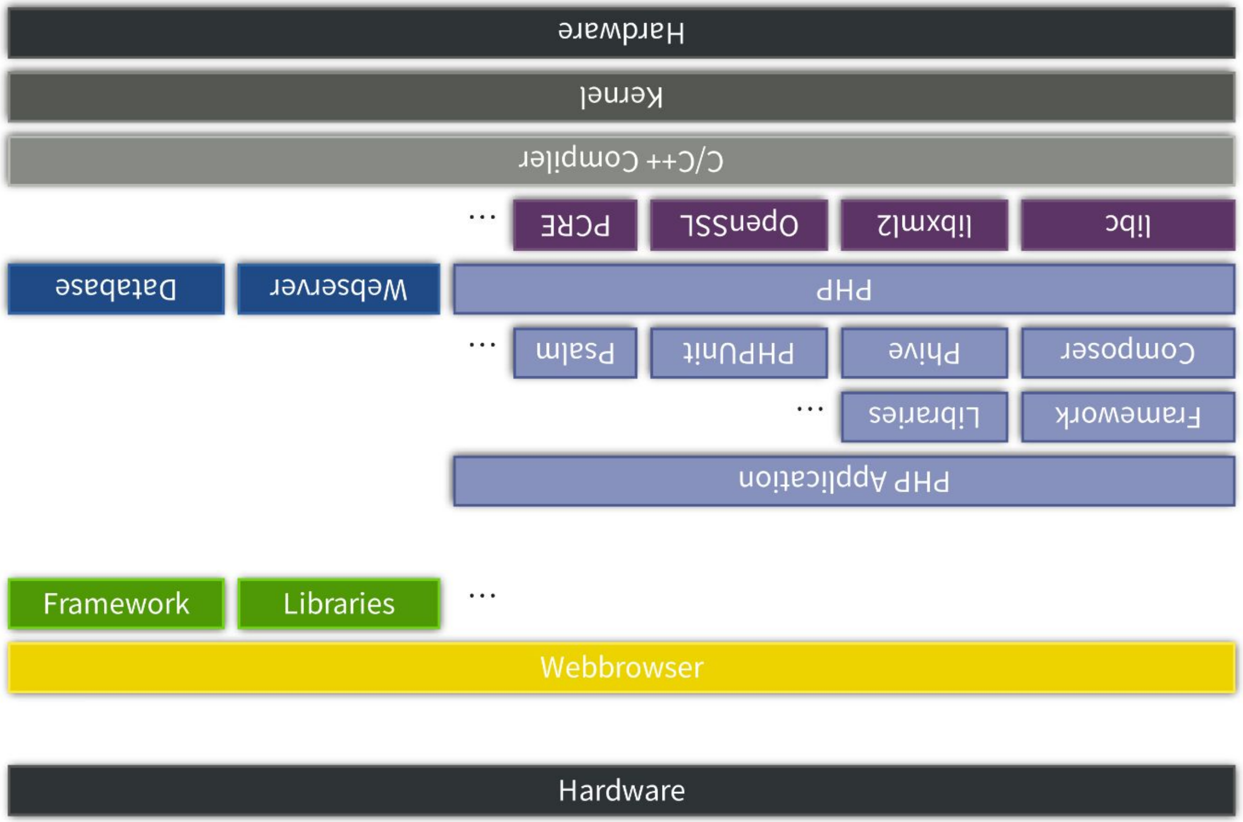
Software Supply Chain

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


Hardware
Kernel
C/C++ Compiler
libc
libxml2
OpenSSL
PCRE
…
PHP
Webserver
Database
Composer
Phive
PHPUnit
Psalm
…
Framework
Libraries
…
PHP Application
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:
    Attacking you through your supply chain

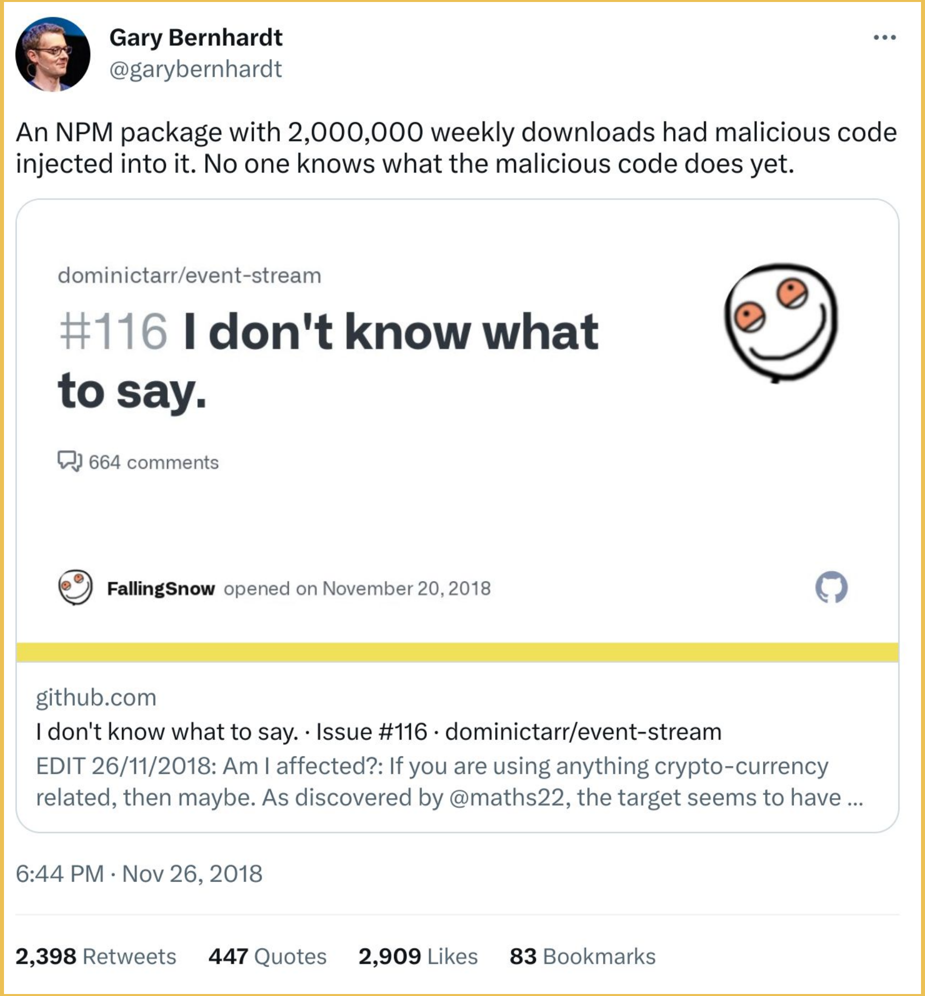
Supply Chain Attacks

- Heartbleed - https://heartbleed.com/ - 2014
  - OpenSSL: System memory accessible externally
- Stuxnet
  - 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
  - discovered in December ‘20 after breach Sep ‘19

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
- XZ Utils / liblzma
  - https://en.wikipedia.org/wiki/XZ_Utils_backdoor
  - Introduced by covert malicious maintainer
  - Backdoor in compression library running in OpenSSH process granting remote access
  - Fortunately detected very early in distribution on March 29th

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

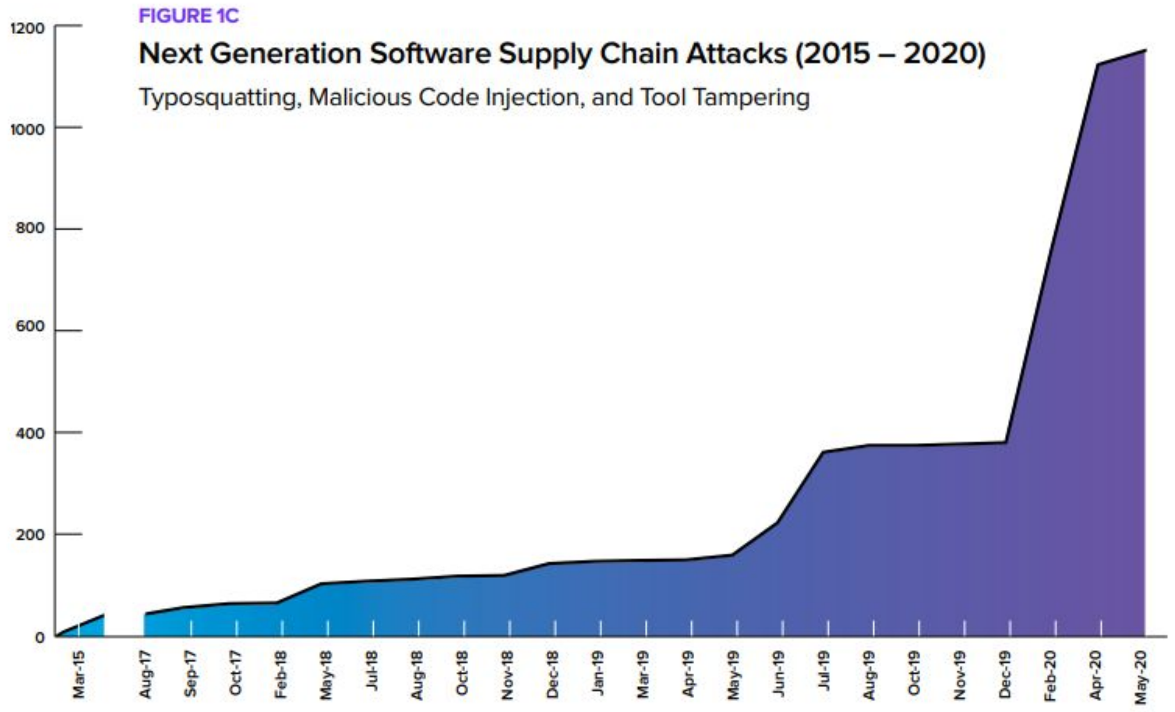
PRIVATE PACKAGIST

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/

PRIVATE PACKAGIST

Supply Chain Attacks



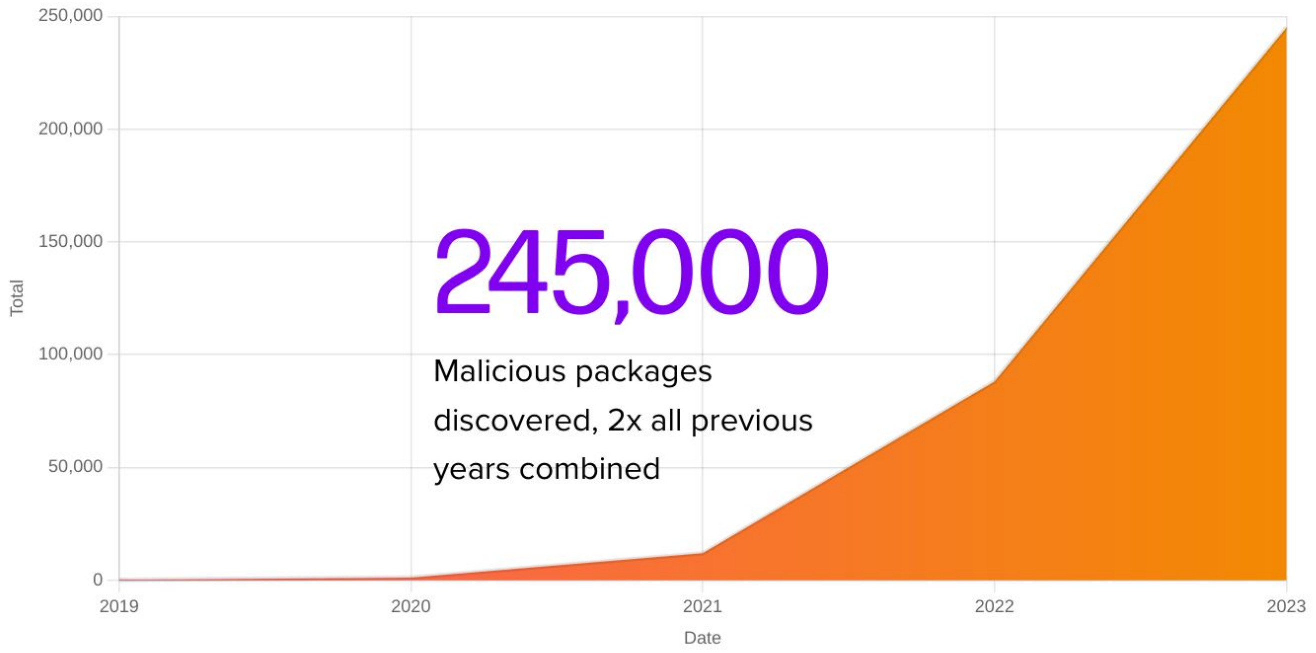
“2020 State of the Software Supply Chain” by sonatype

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

PRIVATE PACKAGIST

Supply Chain Attacks

FIGURE 1.7. NEXT GENERATION SOFTWARE SUPPLY CHAIN ATTACKS (2019-2023)



“9th Annual State of the Software Supply Chain” by sonatype

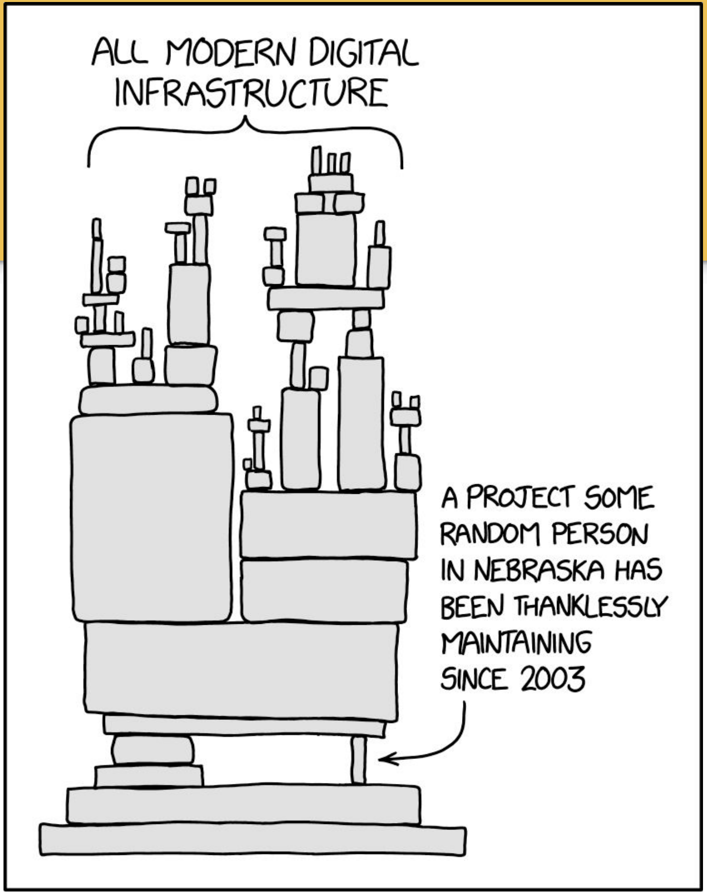
https://www.sonatype.com/hubfs/2023%20Sonatype-%209th%20Annual%20State%20of%20the%20Software%20Supply%20Chain-%20Update.pdf

PRIVATE PACKAGIST

Supply Chain Funding

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

- Creation of Open Software Security Foundation (OpenSSF) at Linux Foundation
  - > $10M raised by 2021
- German Government: Sovereign Tech Fund
  - https://sovereigntechfund.de since 2022
  - €17M budget in 2024, €11.5M in 2023
- Alpha-Omega
  - https://alpha-omega.dev/ since 2022
  - $2.8M granted in 2023


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

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`

Composer Guide to Supply Chain Security

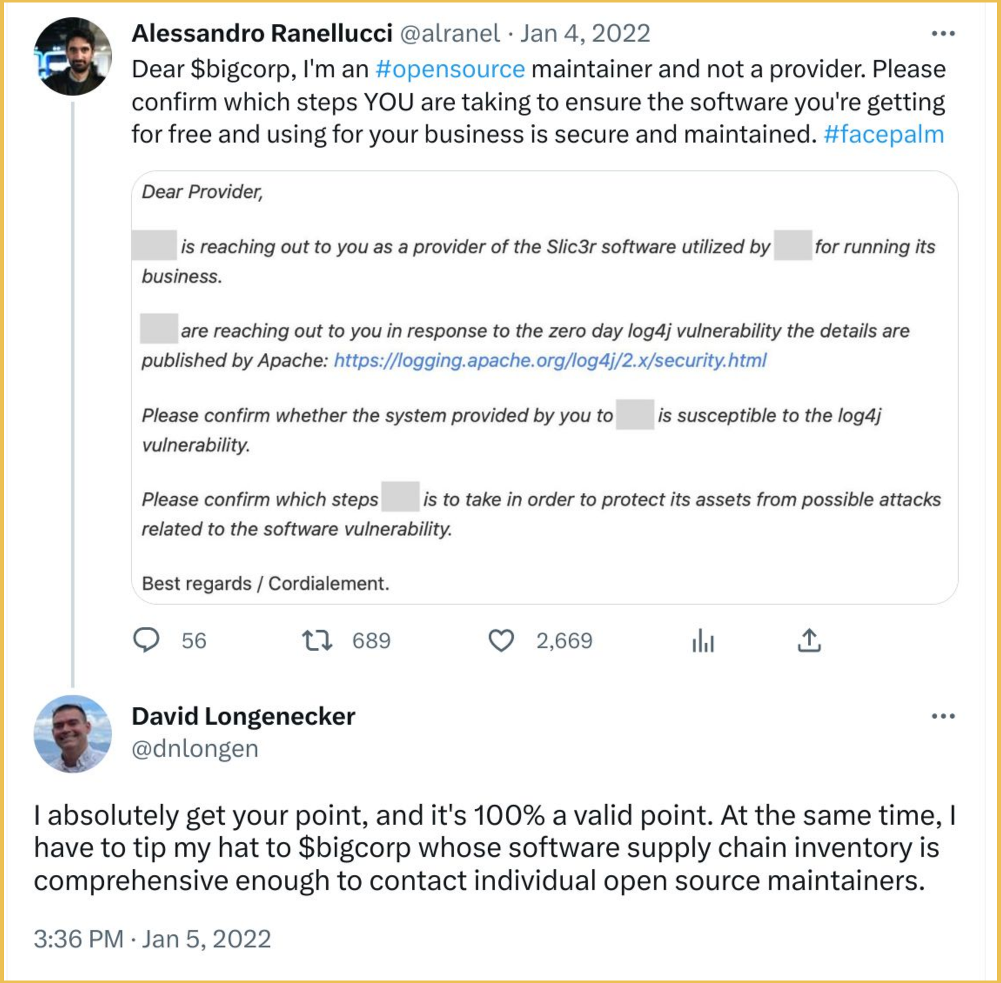
PRIVATE PACKAGIST

Composer Guide: High Level

- 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

PRIVATE PACKAGIST



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.

56 689 2,669

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

PRIVATE PACKAGIST

Composer Guide: High Level

- 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

PRIVATE PACKAGIST

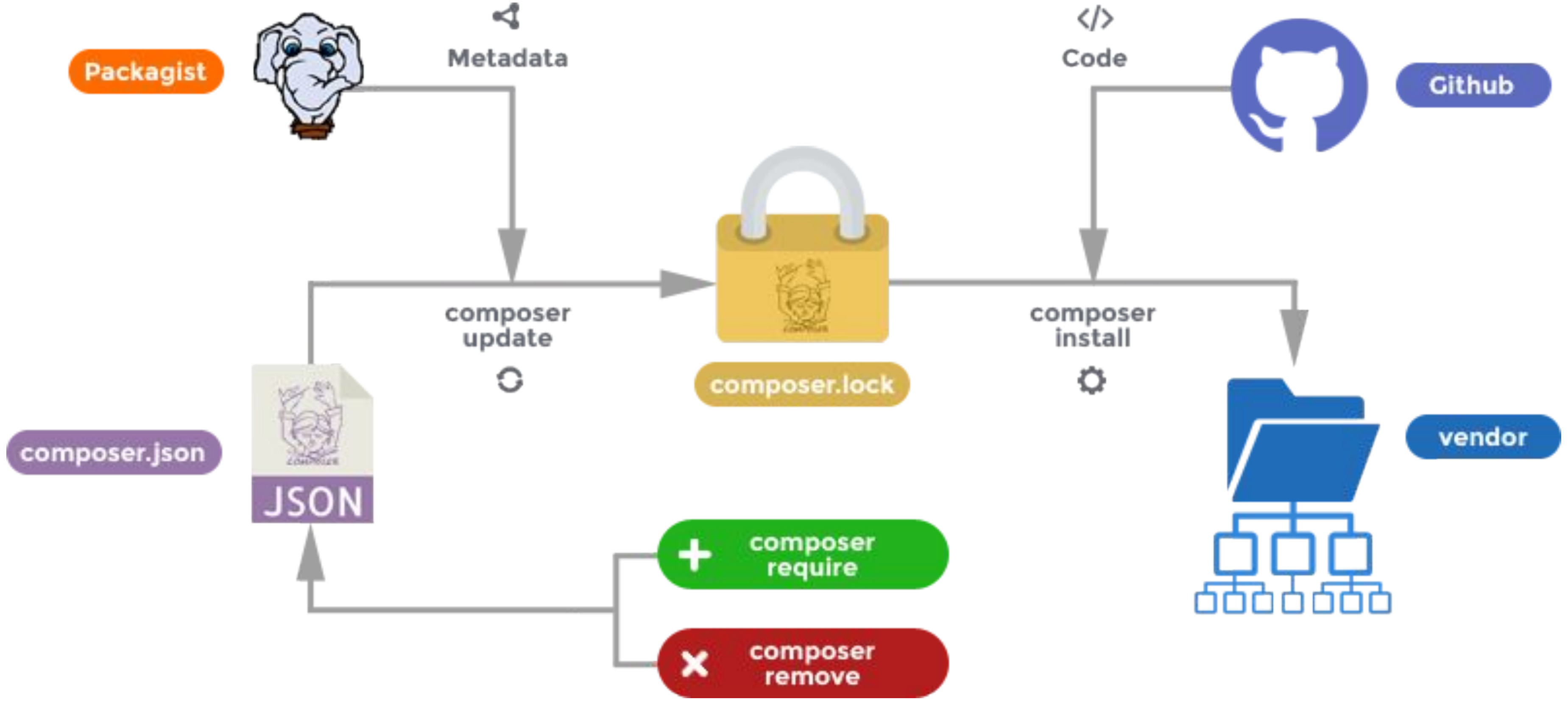
Composer 2.4: composer audit

- **`composer audit`** Command
  - Lists vulnerable versions in composer.lock
  - Uses packagist.org vulnerability db API
    - GitHub advisory database
    - FriendsOfPHP/security-advisories
  - Returns non-zero if vulnerabilities found -> can check in CI

- `composer update implies audit --format=summary`
- `composer require --dev roave/security-advisories:dev-latest`

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 - TUF
    - No way to upload code
- Positively
    - Everything over TLS
    - Installation from GitHub source archive URLs improves trust in artifacts
    - Smaller attack surface on packagist.org

Composer Supply Chain Vulnerabilities

- 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
- Apr 27, 2021: Composer Command Injection Vulnerability
  - https://blog.packagist.com/composer-command-injection-vulnerability/
  - Code execution through Mercurial repository URL injection
- 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

PRIVATE PACKAGIST

Composer Supply Chain Attacks

- May 19, 2022: GitHub Repo Jacking
  - Attacker registered GitHub username of former maintainer
  - Republished package with malicious code to steal AWS credentials
  - https://thehackernews.com/2022/05/pypi-package-ctx-and-php-library-phpass.html
  - https://github.blog/2024-02-21-how-to-stay-safe-from-repo-jacking/
    - Problematic with VCS repo URL references in composer.json too
  - Packagist.org uses GitHub repo ids: https://github.com/composer/packagist/pull/1411
- May 1, 2023: Packagist.org maintainer account takeover
  - https://blog.packagist.com/packagist-org-maintainer-account-takeover/
  - Editing of source URLs no longer allowed beyond 50k installs

PRIVATE PACKAGIST

Protecting yourself from Composer Supply Chain Attacks

- Common wrong suggestion: “Vendoring”
  - Commiting the contents of your vendor directory to source control
- Wrong why?
  - You still need to update your dependencies
    - Either still use the dependency manager to update the vendor’d dependencies
    - Or download everything manually
      - A lot of error prone work
      - Would you notice repo jacking?
  - But there’s more!

PRIVATE PACKAGIST

Why vendoring doesn't protect you

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

Why vendoring doesn't protect you

- 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

PRIVATE PACKAGIST

Why vendoring doesn’t protect you

- 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**

PRIVATE PACKAGIST

Use your own Composer repository

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

- **Private Packagist**

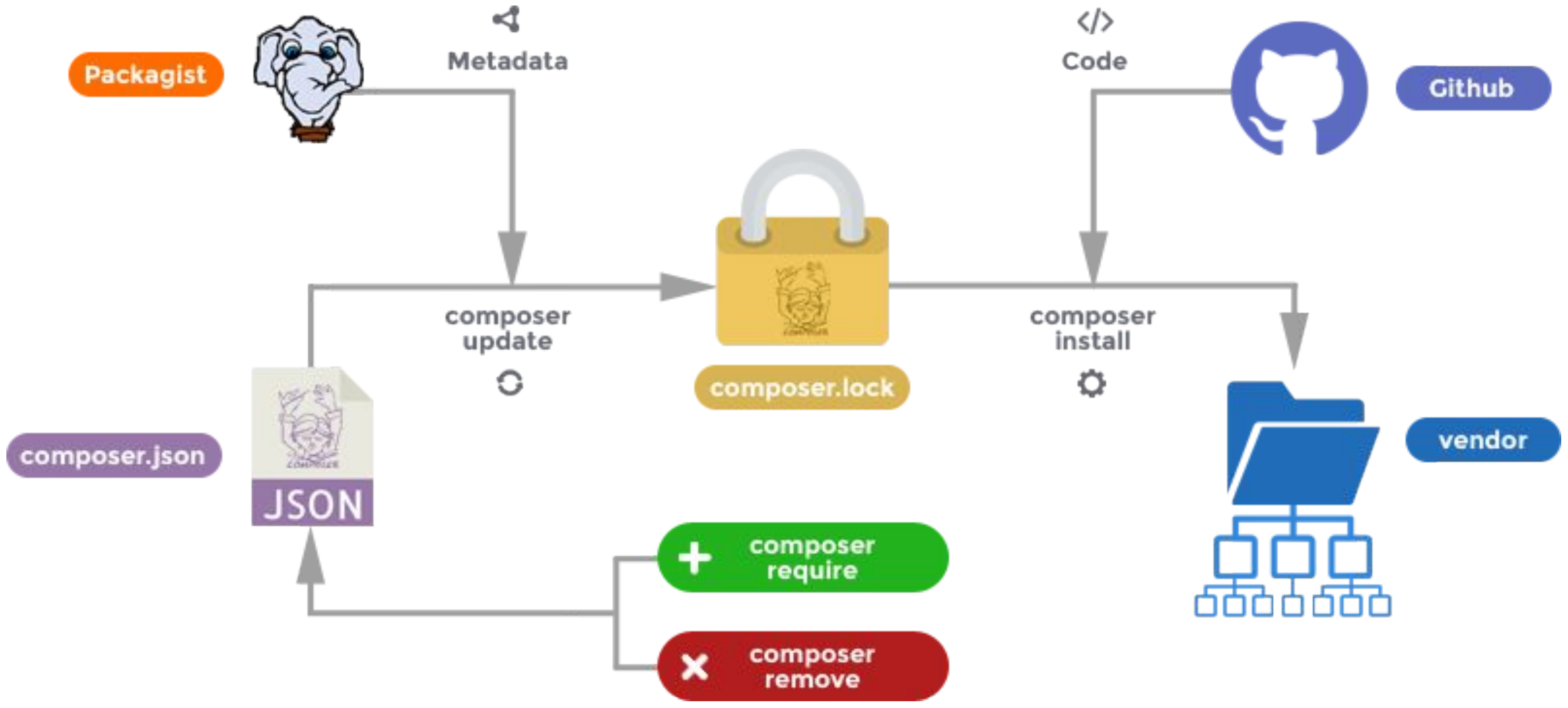
Private Packagist

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

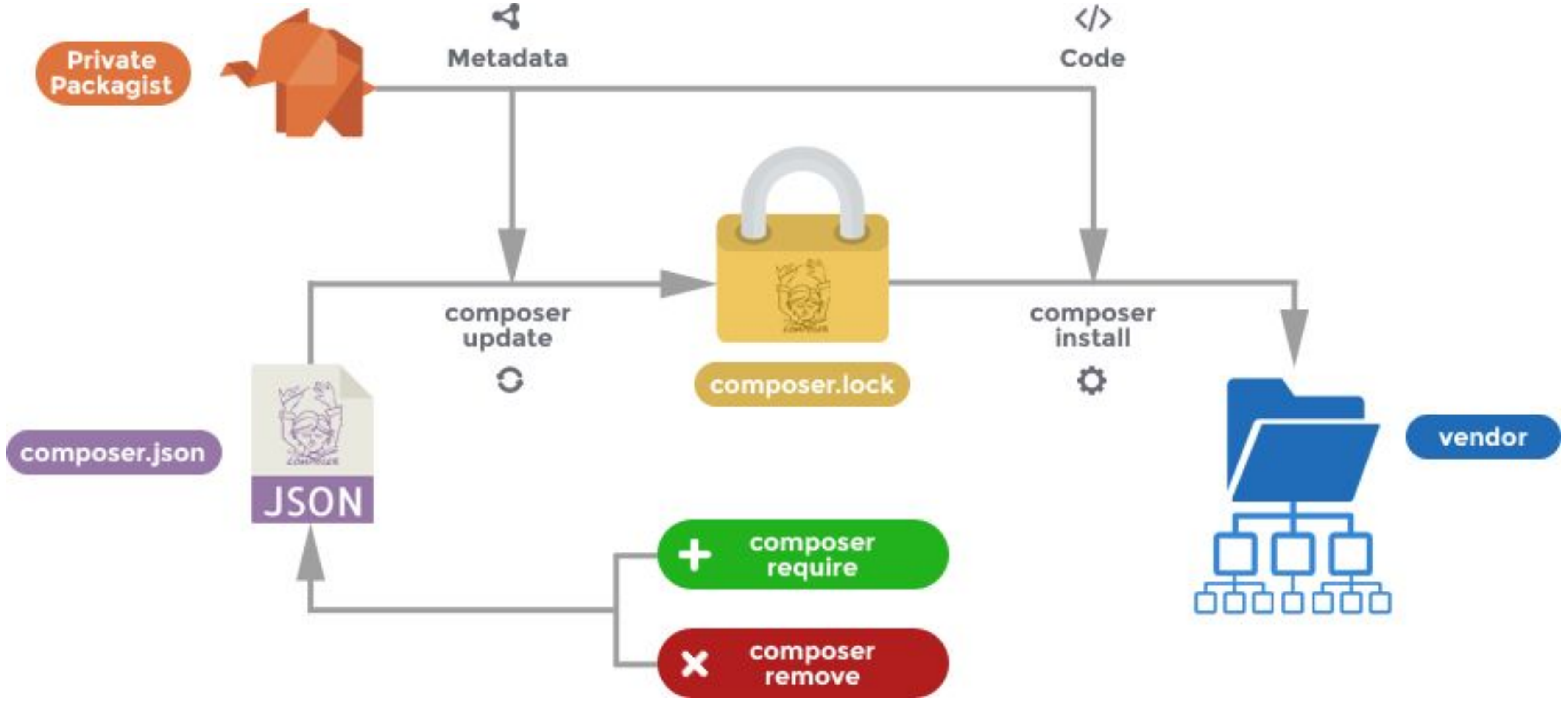
- Serves package metadata **and** code

- 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



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

PRIVATE PACKAGIST

Update Dependencies Frequently

Better yet: Automate your updates

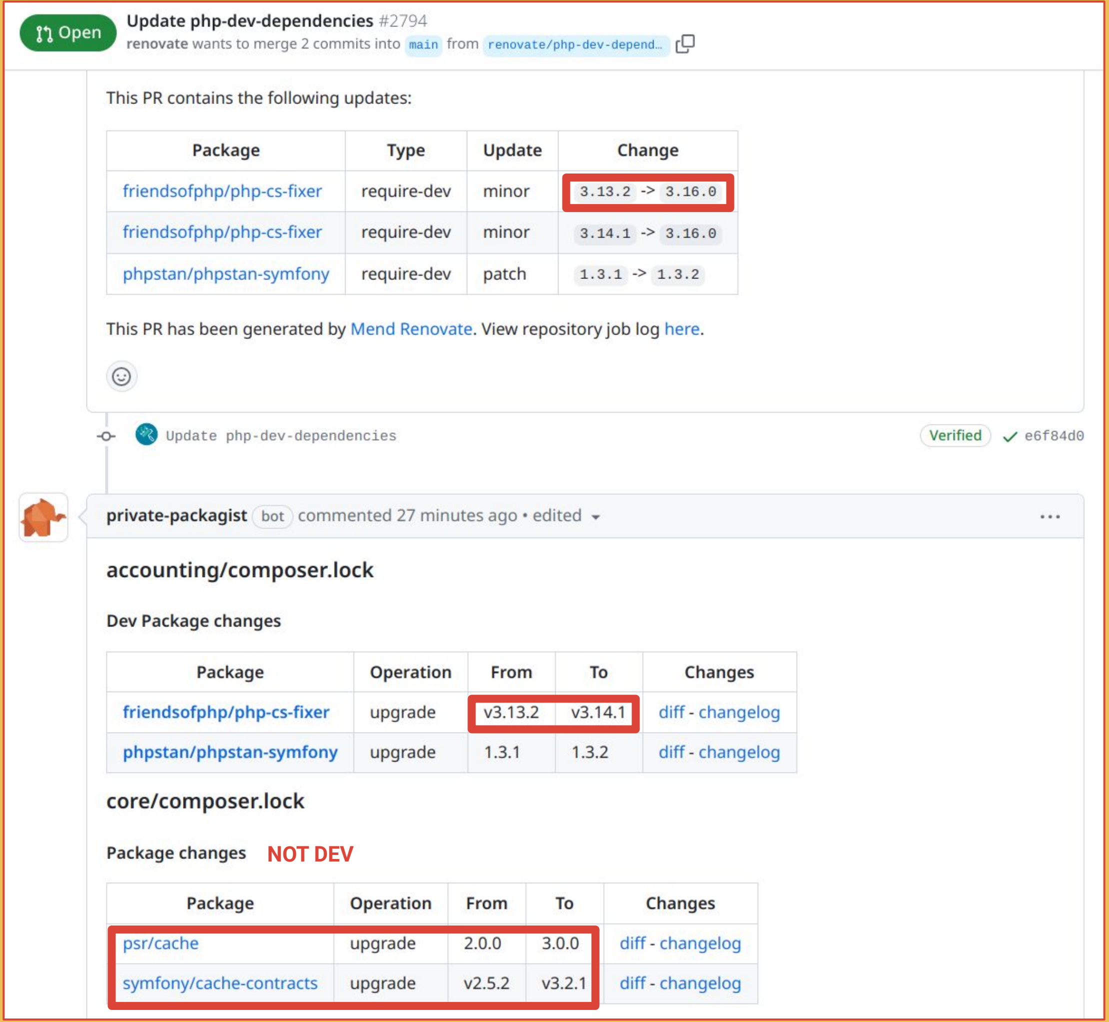
- Mend Renovate https://www.mend.io/renovate/
- GitHub Dependabot https://github.com/dependabot
- *(WIP: Private Packagist Automated Updates)*

Get a pull request anytime an update is necessary

Caution!

Private Packagist Update Review

GitHub
BitBucket
GitLab



Open **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

PRIVATE PACKAGIST

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
  - dump an optimized autoloader
  - package everything into an archive
- deployment
  - upload to production servers, move in place
  - run composer check-platform-reqs
  - 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

PRIVATE PACKAGIST

Composer Guide to Supply Chain Security: Key Takeaways

- composer.lock matters!
  - Commit composer.lock
  - Review changes
- Use a private Composer repository
  - Don't use "Vendoring"
  - Recommendation: Private Packagist
- Automate Dependency Updates
  - Or at least set up monitoring for published vulnerabilities in your dependencies
- Implement a safe deployment process
  - Don't run composer update in deploys

Questions / Feedback?

Private Packagist
https://packagist.com

E-Mail: contact@packagist.com
X: @naderman
Mastodon: @naderman@phpc.social