SYMFONYCON WARSAW 2026

Composer & Packagist Supply Chain Security

Report from the trenches

Nils Adermann & Jordi Boggiano

@naderman · @seldaek · Private Packagist · packagist.com





Composer was born at Symfony events

2010

Symfony Live Paris

Nils and Jordi meet.

2011

SymfonyLive San Francisco

Hackday work on Composer.

2013

SymfonyCon Warsaw

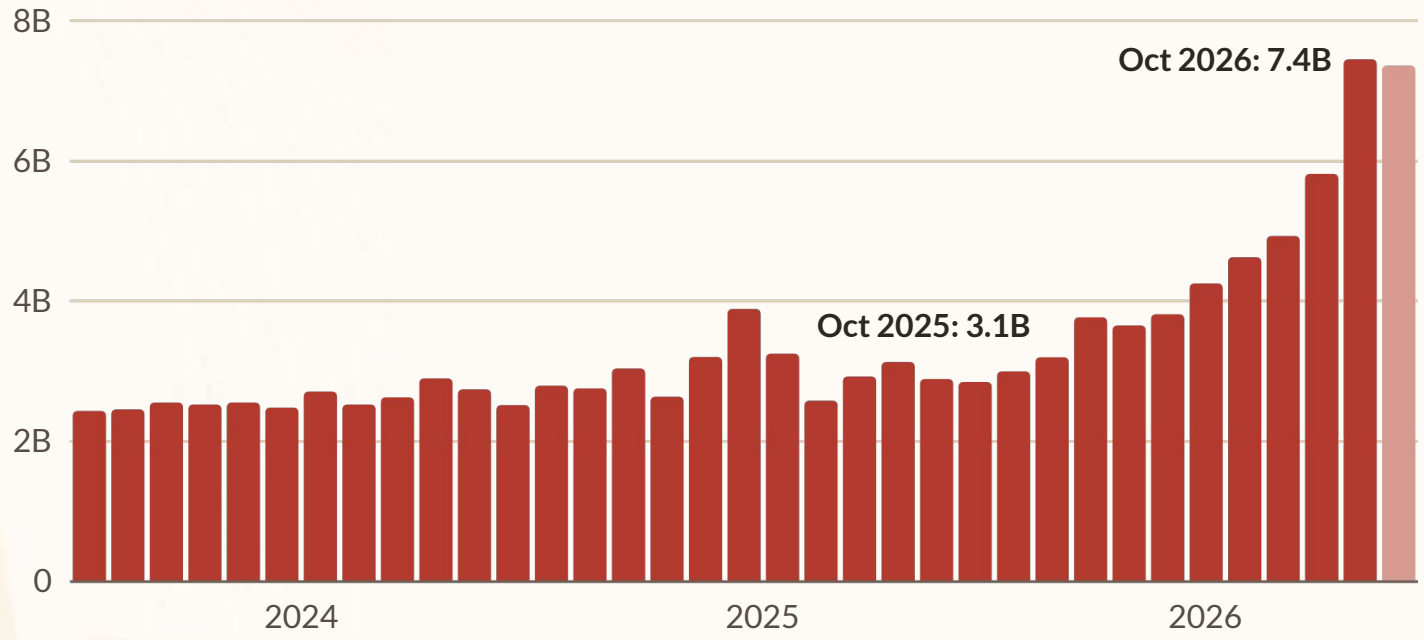
The first SymfonyCon, in this city.

2016

SymfonyCon Berlin

Private Packagist launches and still funds most of Composer and Packagist.org.

Installs per month, January 2024 to November 2026



7.4B

installs in October 2026

2.4× October 2025

51.9B

installs in 2026 so far

36.0B in all of 2025

217B

installs since April 2012

483K packages, 6.1M versions

Lighter bar: November 2026, first 24 days only. Source: packagist.org/statistics, November 24, 2026.

Packagist.org traffic, last month

17B

requests in 30 days
About 650 million on a weekday

6,500

requests per second
Average. +70% since September 2025

37,000

requests per second at peak
Busiest 10 seconds, November 14

76 TB

transferred
JSON metadata and website only, no file downloads

95%

answered by the CDN
Our servers answer the rest in 7 ms (median)

Source: Packagist.org Datadog metrics.

Our plan at SymfonyCon Amsterdam 2025

- **Advisory blocking just added in Composer 2.9**
  composer update skips versions with known security advisories.
- **Transparency log plan just announced**
  A public record of security-relevant events on Packagist.org.
- **2FA enforcement**
  Once organizations exist, later in 2026 or in 2027.
- **Malware blocking planned for Composer 2.10**
  Based on a malware feed from Aikido, integrated into Packagist.org.
- **Organizational package ownership funded**
  Package and vendor ownership by companies and projects, planned for 2026.
- **Artifact integrity scoped, unfunded**
  Stored artifacts, provenance, Trusted Publishing. Funding needed.

01

The trenches

Attacks on PHP packages in 2026.

Attacks and vulnerabilities in 2026

Mar 4	ATTACK	Fake Laravel utilities shipping a remote access trojan
Apr 14	VULN	Composer: Perforce command injection
Apr 30	ATTACK	intercom/intercom-php: existing 5.0.2 tag moved to malicious code
May 13	VULN	Composer: GitHub Actions tokens printed to CI logs
May 22	ATTACK	laravel-lang: about 700 existing tags moved to malicious code
May 23	ATTACK	Eight packages: malware in package.json on dev branches
Jul to Sep	ATTACK	North Korean malware in sevenspan/* and others via hijacked GitHub accounts
Oct 21	VULN	PHP2Shell in PHP, and an out-of-bounds write in the Brotli decoder
Nov 10	ATTACK	CI stealer via 4,837 hijacked GitHub accounts, one Packagist package affected
Nov 10	VULN	Git: code execution during recursive clones
Nov 18	VULN	nginx2root: HTTP/2 vulnerabilities in nginx
Nov 24	VULN	Composer 2.11.3: code execution in a directory prepared by an attacker

How a release gets from GitHub to your project

GitHub repository
Versions are git tags, Push access is enough to publish.

→

Packagist.org
Stores: commit hash, GitHub zip URL

→

Composer
Downloads zip, runs plugins

⚠ **Until June 2026, Packagist.org updated a version when its tag was moved to another commit.**

- Every install can be traced back to a commit in the source repository.
- GitHub builds the zip on request: no stored file to checksum or sign.
- composer install kept the locked commit. composer update picked up the moved tag.

intercom/intercom-php: malicious code under version 5.0.2

ATTACK

1. The attacker gained GitHub repository access (“Mini Shai-Hulud” campaign).
2. The attacker force-pushed the existing 5.0.2 tag to a malicious commit.
3. The new composer.json declared the package as a Composer plugin.
4. Once the allow-plugin prompt was accepted, the plugin stole cloud credentials, .env files and SSH keys.
5. Aikido’s malware feed flagged the version. We removed it within minutes.

WHAT MADE THE ATTACK POSSIBLE

- ⚠ Packagist.org accepted the moved 5.0.2 tag and served the new commit.
- ⚠ Composer had no malware blocking. Users relied on our quick removal.
- ⚠ The new plugin prompt is easy to accept for a package already in use.

laravel-lang: about 700 existing tags moved

ATTACK

1. The attacker gained GitHub access with a stolen token.
2. The attacker moved about 700 existing tags of four packages to malicious commits, seconds apart.
3. src/helpers.php was added to autoload.files, executed when the app loads the autoloader.
4. It stole cloud and Kubernetes credentials, SSH keys, .env files, browser data, wallets.
5. Aikido flagged the versions. We removed them within minutes and unlisted the packages during cleanup.

WHAT MADE THE ATTACK POSSIBLE

- ⚠ Packagist.org accepted moved tags for every existing version.
- ⚠ composer.lock only protects until the next composer update.

WHAT STILL HELPED

- ✓ allow-plugins: no attacker code ran during install.
- ✓ Reviewing the composer.lock diff: existing versions showed new commit hashes.

Smaller attacks: malware in JavaScript build files

- **May 23: eight packages**
  postinstall script in package.json on dev branches. Downloads and runs a Linux binary.
- **June to November: many small attacks**
  Little-used or attacker-created packages. Malware in frontend build files, for example font files.
- **November: 4,837 hijacked GitHub accounts**
  Stealer in CI workflows and npm postinstall hooks. Reached digitalelvis/neuronai-studio 3.14.3 through automated publishing.

LESSONS

- ⚠ Composer doesn't run npm scripts, but npm install and your frontend build do.
- ⚠ Scanning only composer.json misses this malware.
- ✓ Dev versions follow branches, so their code can change at any time. Deploy tagged releases.

What the 2026 attacks have in common

- **GitHub push access is enough to publish**
  Entry points: JS dependencies, IDE plugins, tools installed by AI agents, misconfigured GitHub Actions.
- **Malware tried plugins, and autoload to skip user interaction**
  autoload.files runs when the app loads the autoloader. npm scripts and CI workflows run in the build.
- **Existing versions were changed**
  Same version number, different code. composer.lock only protects until the next update.
- **More attacks, across all registries**
  npm and PyPI now link packages to their source, as Composer does with git tags. AI tools make attacks cheaper.

02

Phase 1: Detection

Find malicious versions fast, remove them fast, and keep a public record of every change.

Malware detection and incident response

- **Aikido malware feed in Packagist.org**
  Imported since March 2026. Flagged versions are marked on the site and in the metadata Composer reads.
- **Reports from Socket**
  Through private channels. More providers welcome.
- **Fast removal by the Packagist team**
  Popular packages cleaned within minutes, at any hour.
- **New admin tools**
  Flag and unflag, soft deletion with a reason, freezing packages and accounts.

THANK YOU

- Aikido: malware feed and sponsorship
- Socket: reports and sponsorship

Transparency log for Packagist.org

Every moved tag in the 2026 attacks is in the log, with exact timestamps.

RECORDED EVENTS

- Maintainers added or removed, transfers
- Packages abandoned, deleted or frozen
- The commit of each version, blocked tag moves
- Password resets, email and username changes
- Changes to advisories and the malware list

ACCESS

- packagist.org/transparency-log, after login
- Filter by event, user, vendor, package or date

```
?package=symfony/console
```

NEXT

- 2FA events and 2FA status, in the coming weeks
- A read API
- A tamper-evident design

03

Phase 2: Prevention

Make the safe outcome the default, in Composer and on Packagist.org.

Composer 2.10: dependency policies

- **A single config.policy section**
  Malware, advisories and abandoned packages. Replaces config.audit.
- **The same three settings for every policy**
  block: remove from resolution. audit: ignore, report or fail. ignore: exemptions with a reason.
- **Custom policies**
  Your own list of disallowed versions, from a URL or a Composer repository.
- **Temporary override**
  --no-blocking, or COMPOSER_NO_BLOCKING in CI, for temporary relief.

```
$ composer install
Installing dependencies from lock file
Your lock file does not contain a compatible set of
packages.

  Problem 1
    - Package evil/badpkg v1.0.0 (in the lock file) was not
loaded, because it was flagged as malware
    reason: malware. To ignore filters for this package,
add it to "policy.malware.ignore". To turn the feature off,
set "policy.malware.block" to false.
```

Default dependency policies in Composer 2.10

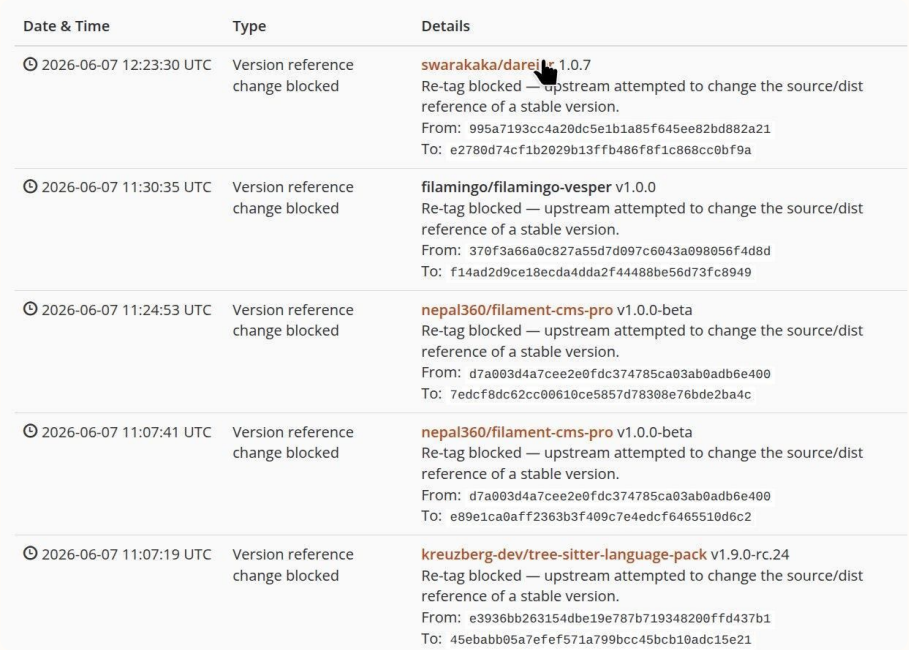
Policy	update, require	install from lock	audit
Malware	Blocked	Blocked	Fails
Advisories	Blocked	Installs, to assess impact	Fails
Abandoned	Allowed	Allowed	Reported

- **Malware is checked on install too**
  A version flagged is refused even on install so malware never makes it to disk.
- **Custom policies use the same settings**
  Same block, audit and ignore settings. block-scope “all” also checks on install (2.11).

```
{
  "config": {
    "policy": {
      "company-policy": {
        "sources": [{
          "type": "url",
          "url": "https://acme.test/bad.json"
        }],
        "audit": "fail",
        "block-scope": "all"
      }
    }
  }
}
```

Packagist.org: stable versions are now immutable

- **Re-tagging a stable version is rejected**
  The version keeps its commit. Badge on the package page, email to all maintainers, entry in the log.
- **dev- versions still follow their branches**
  A branch moves with every push.
- **Deleting is now a soft delete, with a reason**
  Never installable again. Recoverable where appropriate. Transparent to users.
- **Made a mistake? Tag the next version**
  3.2.0.1 works: Composer supports four version components.



Date & Time	Type	Details
2026-06-07 12:23:30 UTC	Version reference change blocked	swarakaka/dareik v1.0.7 Re-tag blocked — upstream attempted to change the source/dist reference of a stable version. From: `995a7193cc4a20dc5e1b1a85f645ee82bd882a21` To: `e2780d74cf1b2029b13ffb486f8f1c868cc0bf9a`
2026-06-07 11:30:35 UTC	Version reference change blocked	**filamingo/filamingo-vesper** v1.0.0 Re-tag blocked — upstream attempted to change the source/dist reference of a stable version. From: `370f3a66a0c827a55d7d097c6043a098056f4d8d` To: `f14ad2d9ce18ecda4dda2f44488be56d73fc8949`
2026-06-07 11:24:53 UTC	Version reference change blocked	nepal360/filament-cms-pro v1.0.0-beta Re-tag blocked — upstream attempted to change the source/dist reference of a stable version. From: `d7a003d4a7cee2e0fdc374785ca03ab0adb6e400` To: `7edcf8dc62cc00610ce5857d78308e76bde2ba4c`
2026-06-07 11:07:41 UTC	Version reference change blocked	nepal360/filament-cms-pro v1.0.0-beta Re-tag blocked — upstream attempted to change the source/dist reference of a stable version. From: `d7a003d4a7cee2e0fdc374785ca03ab0adb6e400` To: `e89e1ca0aff2363b3f409c7e4edcf6465510d6c2`
2026-06-07 11:07:19 UTC	Version reference change blocked	kreuzberg-dev/tree-sitter-language-pack v1.9.0-rc.24 Re-tag blocked — upstream attempted to change the source/dist reference of a stable version. From: `e3936bb263154dbe19e787b719348200ffd437b1` To: `45ebabb05a7efef571a799bcc45bcb10adc15e21`

Re-tag attempts, blocked and logged.

Composer 2.11: cooldown policy for new releases

- **Holds back versions in the cooldown period**
  Applies to update and require. Not to install, not to dev versions.
- **Publication time set by Packagist.org**
  published-time: when Packagist.org first served the version. Publishers can't set it.
- **No automatic exemption for security fixes**
  A fake advisory could skip it. Use --no-cooldown per package.

- **Recommended: a few hours to 1 day**
  Our best balance between:
  - Security fixes with only a short delay, even if you don't monitor advisories
  - Automatic malware detection completing before you install

```
{
  "config": {
    "policy": {
      "cooldown": {
        "period": "1 day",
        "ignore": {
          "acme/*": "Our own packages"
        }
      }
    }
  }
}

$ composer update --no-cooldown vendor/pkg
```

04

Accounts and ownership

Taken-over accounts and stolen tokens started the 2026 attacks. 2FA, organizations and account protection on Packagist.org.

2FA roadmap for Packagist.org accounts

COMING WEEKS

Security keys

FIDO2 hardware keys andd passkeys as second factor. Ships first.

COMING WEEKS

Public 2FA status

2FA events in the transparency log and 2FA status on profiles, together.

PLANNED

Advanced Account Protection

Opt-in: sensitive actions need a recent security key check. Details in progress.

LATER

Mandatory 2FA

For all accounts. Stricter for organizations and popular packages. With lead time.

For comparison: PyPI has required 2FA since January 2024. npm is moving to FIDO-only 2FA.

Organizations on Packagist.org

Members use their own accounts and their own 2FA. No more shared company logins.

In testing since July and August

- Organizations, created by our staff for now
- Two fixed teams: owners and all members
- Owners must use 2FA
- Invite people by email into teams
- Owners see their organization's full history

Finishing: membership policies

- Require 2FA for every member
- Require an email on an approved domain
- Require membership in a GitHub organization
- Members who stop complying are suspended, not removed

Next

- Vendor prefixes: control who publishes under acme/
- Package ownership: move packages into the organization
- Access recovery when no owner can act
- Then open to companies and open source projects on request

FROM OUR SPONSOR OUTREACH

Many company packages belong to accounts with personal emails of former employees. The company controls the GitHub repository, but not the Packagist.org package.

05

Hardening our own stack

Composer itself, the servers behind Packagist.org, and our CI.

Composer security fixes, November 2025 to November 2026

15

security advisories

Each published as a GitHub security advisory

27

further hardening fixes

Listed as security changes, too small for an advisory

14

releases

From 2.9.3 to 2.11.3, plus 3 release candidates

EXAMPLES

- Perforce command injection via package metadata (2.9.6)
- GitHub Actions tokens printed to CI logs (2.9.8)
- Package names and bin paths escaping vendor/ (2.10.2 to 2.11.0)
- Credentials sent to other hosts through a crafted composer.lock (2.11.2)
- Zip archives writing outside the package directory via symlinks (2.11.2)
- Code execution in an attacker-prepared directory, bypassing the CVE-2024-24821 fix (2.11.3, this week)

“Black October”: vulnerabilities in the Packagist stack

- **PHP2Shell, October 21**
  Two unauthenticated RCEs in PHP, exploited in the wild
- **Brotli decoder, October 21**
  Out-of-bounds write, fix public for eight months
- **Git, November 10**
  Six vulnerabilities, including code execution on recursive clone
- **nginx2root, November 18**
  Five nginx vulnerabilities, two of them in HTTP/2

OUR RESPONSE

- Packagist.org is critical infrastructure for the PHP ecosystem
- Safe from each vulnerability early on
- Own builds instead of distribution packages in places, for early patches
- Hardening and attack surface reduction
- More disclosures expected

RECOMMENDED TALK

Volker Dusch: “Fixing PHP2Shell”, Friday at 11:00

Hardening our own GitHub Actions

- **Pin every action to a full commit SHA**
  Moved tags of tj-actions/changed-files hit 23,000+ repositories in 2025. Dependabot or Renovate update the pins.
- **Least privilege**
  permissions: {} by default, grant per job. persist-credentials: false on checkout.
- **Safe workflow triggers**
  No pull_request_target with a fork checkout. Treat ${{ }} with untrusted input as code. Blocked by GitHub from Dec 2nd, 2026.
- **zizmor as a blocking check**
  Pedantic mode, on every workflow change and every maintained branch.

```yaml
permissions:
  contents: read
# …
steps:
  - uses: actions/checkout@<sha> # v7.0.0
    with:
      persist-credentials: false
  - uses: zizmorcore/zizmor-action@<sha>
    with:
      persona: pedantic
      advanced-security: false
```

zizmorify, a Claude Code skill that applies these changes: github.com/Seldaek/zizmorify

GitHub now has lock files for actions, in public preview: docs.github.com/en/actions/concepts/security/workflow-dependency-lockfiles

06

Recommendations

Steps for your Symfony apps and for the packages you maintain.

Recommendations for Symfony apps

1. composer self-update to 2.11: laptops, CI, Docker images, agent sandboxes.
2. Set a cooldown of a few hours to a day. --no-cooldown for urgent fixes.
3. Keep policies on. Ignore narrowly, with a reason, never "policy": false.
4. Commit composer.lock, review its diffs, deploy with composer install.
5. Carefully select individual plugins you trust (allow-plugins)
6. Require tagged releases, not dev-main.

   Don't set minimum-stability. If you need a dev version, allow it for that package, not the whole project: "acme/lib": "^2.1@dev"

```
{
  "config": {
    "allow-plugins": {
      "php-http/discovery": true,
      "symfony/flex": true,
      "symfony/runtime": true
    },
    "policy": {
      "cooldown": { "period": "1 day" }
    }
  }
}
```

Recommendations for package maintainers

1. 2FA on Packagist.org and GitHub now. Add a security key when FIDO2 ships.
2. Never re-tag. Release the next version instead: 3.2.0.1 is fine.
3. Personal accounts, no shared logins. Move to an organization once available.
4. GitHub push access is enough to publish. Use fine-grained, short-lived tokens.
5. Harden your workflows with zizmor or zizmorify: pinned SHAs, permissions: {}.
6. Watch your packages in the transparency log, and read the “retag blocked” emails.

IF YOUR PACKAGE IS COMPROMISED

- Revoke GitHub tokens and sessions. Rotate CI secrets.
- Contact us. We can freeze the package during cleanup.
- Check the transparency log for changed versions.
- Release fixed versions. Tell users which to avoid.

Private Packagist: supply chain security for PHP

Composer and Packagist.org protections, enforced for the whole organization.

SUPPLY CHAIN SECURITY SETTINGS

MALWARE AND POLICIES

- Malware downloads blocked for every Composer version
- Org-wide policies and cooldown

COMPOSER CLIENT CONTROLS

- Required Composer version: 2.2 LTS or latest
- Plugin allowlist by name pattern

DOWNLOAD PATHS AND ACCESS

- No upstream dist or source URLs

DEVELOPMENT SECURITY FEATURES

- **Security Monitoring**
  Alerts by email, Slack, Teams or webhook when an advisory affects your projects.
- **Trusted Publishing**
  Publish your own artifact packages from CI, without long-lived secrets.
- **Update Review**
  Every composer.lock change in a pull request comment, with malware and advisories flagged.
- **Package Usage Tracking**
  Which applications use a package, and in which version.

07

2027 and beyond

Status of last year’s plan, verifiable artifacts on Packagist.org, the remaining roadmap, and funding.

Artifact integrity for Packagist.org and Composer

GOAL

Immutability for artifacts, not just version metadata, and verifiable origin for every artifact.

PHASE 1

Artifact storage

Zstd archives of all tagged versions with checksums.

PHASE 2

Signatures and provenance

Sigstore signatures and build provenance, verified by Composer.

PHASE 3

Scoped API tokens

Per package and permission, with expiry. Revoked if leaked.

PHASE 4

Trusted Publishing

Chosen CI workflows publish without long-lived secrets.

PHASE 5

composer publish

Sign, upload and stage releases. Approve with a hardware key.

Uploads and Trusted Publishing launch only with mandatory signing and attestations.

Quarterly progress updates on blog.packagist.com.

Artifact integrity: benefits and limits

- **Verifiable installs**
  Checksums for all stable versions. SHA-256, signatures and attestations in new Composer releases.
- **Monorepos without subtree splits**
  Several packages from a single repository. Finally separate AWS SDK packages!
- **Build steps without a second repository**
  PHARs like PHPStan's, published from the source repository.
- **Mirror and private package registry contents you can verify**
  Compare mirrored copies with what was published.

Only protects transport

- ⚠ A taken-over maintainer account still signs validly. Staged releases with a hardware key reduce the risk.
- ⚠ Code review of your dependencies remains necessary.

Roadmap for 2027, in order of work

In progress

- Security keys, then public 2FA events and 2FA status on profiles
- Membership policies for organizations
- Artifact storage, with checksums in metadata and the log

Next

- Advanced Account Protection
- Vendor prefixes and package ownership, then organizations for everyone
- A transparency log API, then a tamper-evident log
- Signatures, provenance, scoped tokens, Trusted Publishing
- Staged releases, approved with a security key
- Mandatory 2FA / hardware security keys

Needs funding

- Verification and selection policies in Composer
- A hardened build environment for packages without Trusted Publishing
- More hardening of our own infrastructure and processes
- composer vet (see cargo vet)

Funding Composer and Packagist.org

~$1M

2026 expenses
To run the existing services

~$1.4M

needed
For the work on this roadmap

~$500K

from Private Packagist
About half of the total

~$500K

from sponsors
Program launched in July

SPONSORSHIP TIERS

- Bronze: €2,500 per month
- Silver: €5,000 per month
- Gold: €8,000 per month
- Annual commitment

SPONSORS

Private Packagist, Aikido, AWS, Socket, Bunny, Upsun, Sonatype, Sentry, Tideways, Datadog, Algolia

SECURITY WORK

- Paid by grants so far
- Sovereign Tech Agency until September
- Alpha-Omega since November
- Needs regular funding

Sponsor us: packagist.org/sponsor · sponsoring@packagist.org

Find us at SymfonyCon

Nils Adermann
Bluesky @naderman.de
Mastodon @naderman@phpc.social
LinkedIn in/nilsadermann
X @naderman

Jordi Boggiano
Bluesky @seld.be
GitHub @Seldaek
LinkedIn in/seldaek
X @seldaek

Also here from the team and contributors

Stephan Vock · Eirik Morland · Igor Benko
Steven Rombauts · Yanick Witschi

packagist.com · packagist.org/sponsor · contact@packagist.com



Celebrating 15 years of Composer