



WHAT IS THE
SECURE SOFTWARE SUPPLY CHAIN
AND THE CURRENT STATE OF THE
PHP ECOSYSTEM





Paolo Mainardi



@paolomainardi

- Co-founder and CTO @[Sparkfabrik](#)
- [Linux Foundation Europe Advisory Member](#)
- Blog: [paolomainardi.com](#)
- Podcast: [Continuous Delivery](#)
- [linkedin.com/in/paolomainardi](#)
- [continuousdelivery.social/@paolomainardi](#)
- pao.lo.mainardi@sparkfabrik.com



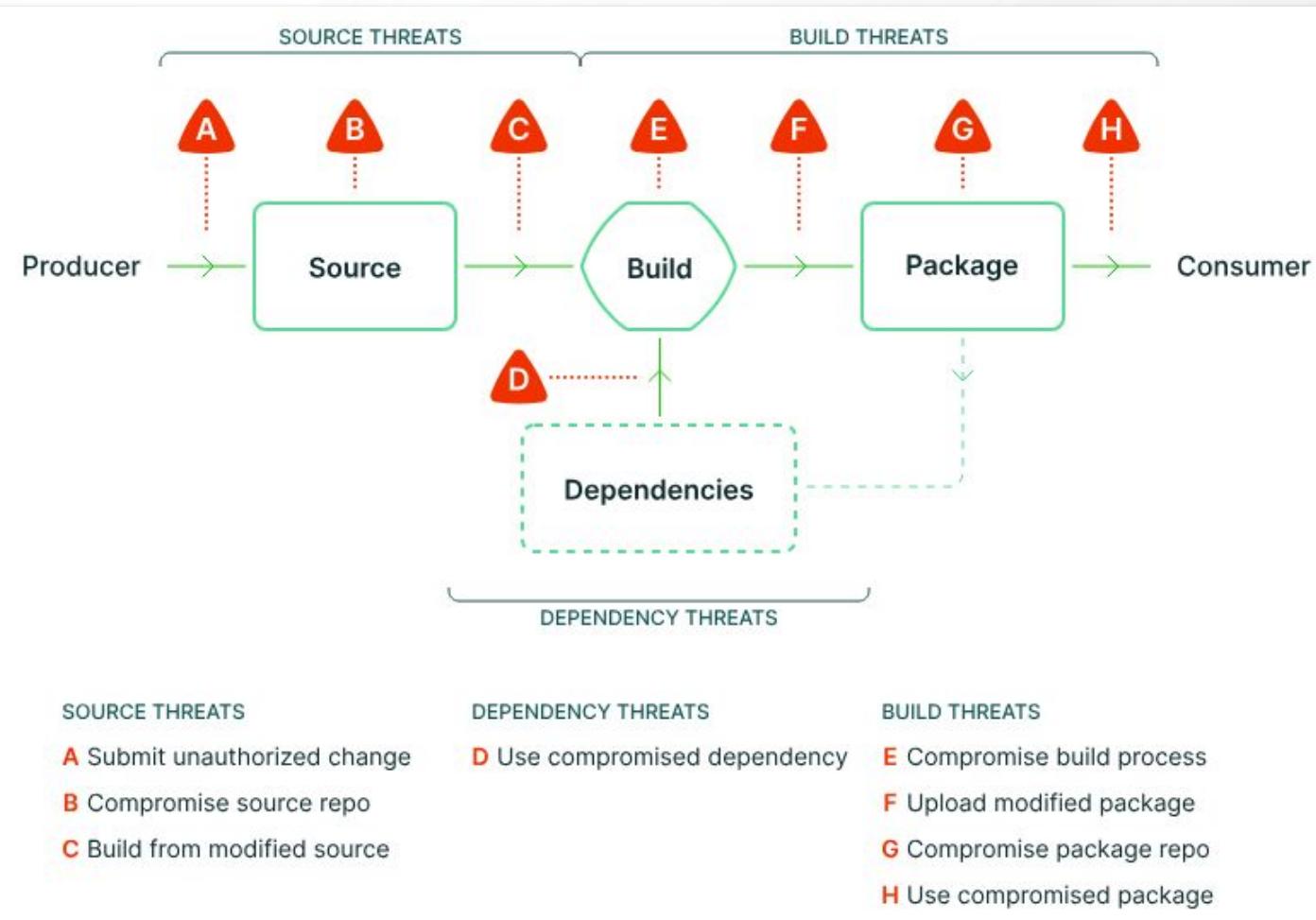
THE SESSION

- What is a **Software Supply Chain**
- State of the **PHP** ecosystem
- What are **the threats** and how can be challenged with:
Sigstore - SBOM - OSV - Scorecard



A **supply chain** is a network of individuals and companies
who are involved in creating a product
and delivering it to the consumer





A MODERN PHP APPLICATION

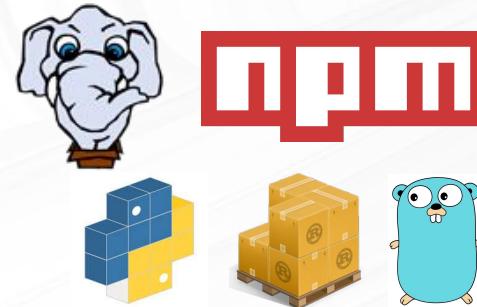


A MODERN PHP APPLICATION

Application



Dependencies



A MODERN PHP APPLICATION

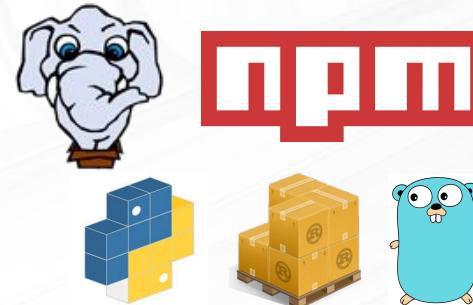
Operating system



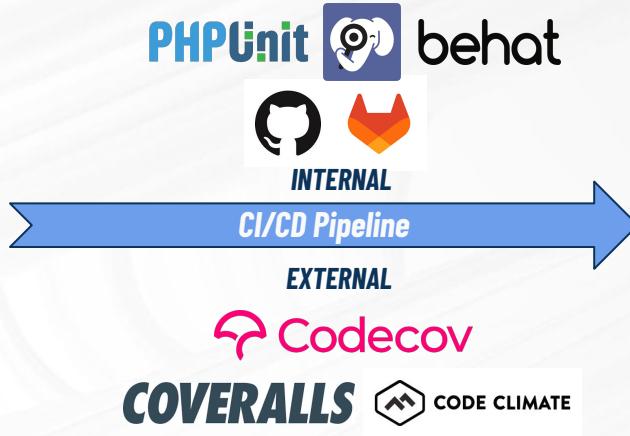
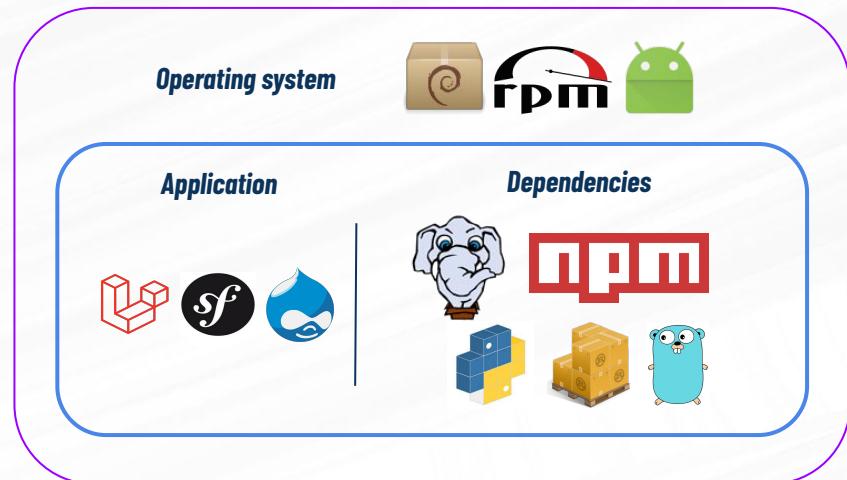
Application



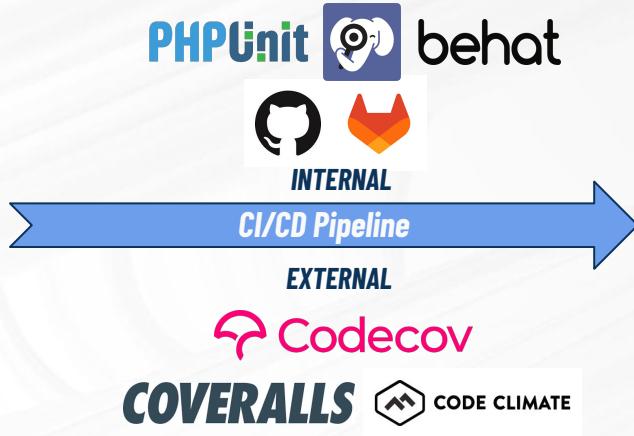
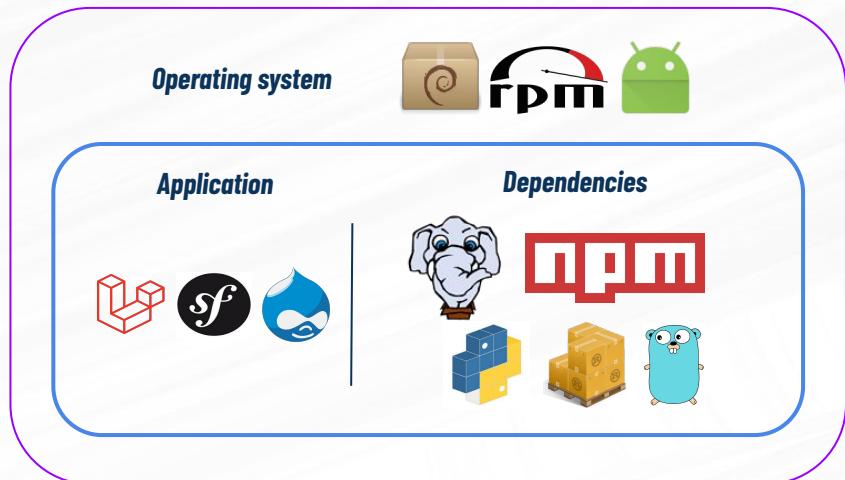
Dependencies



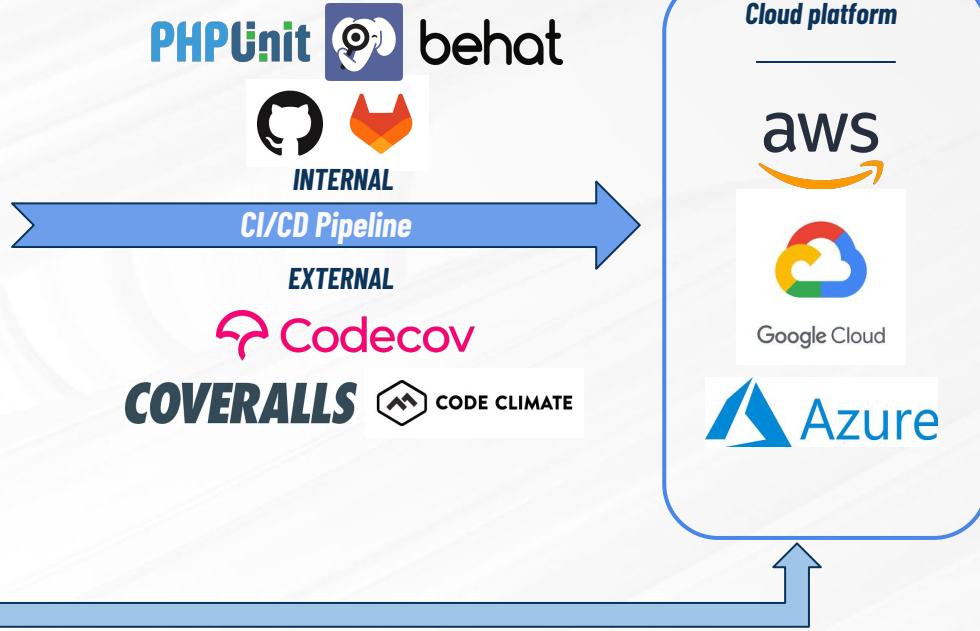
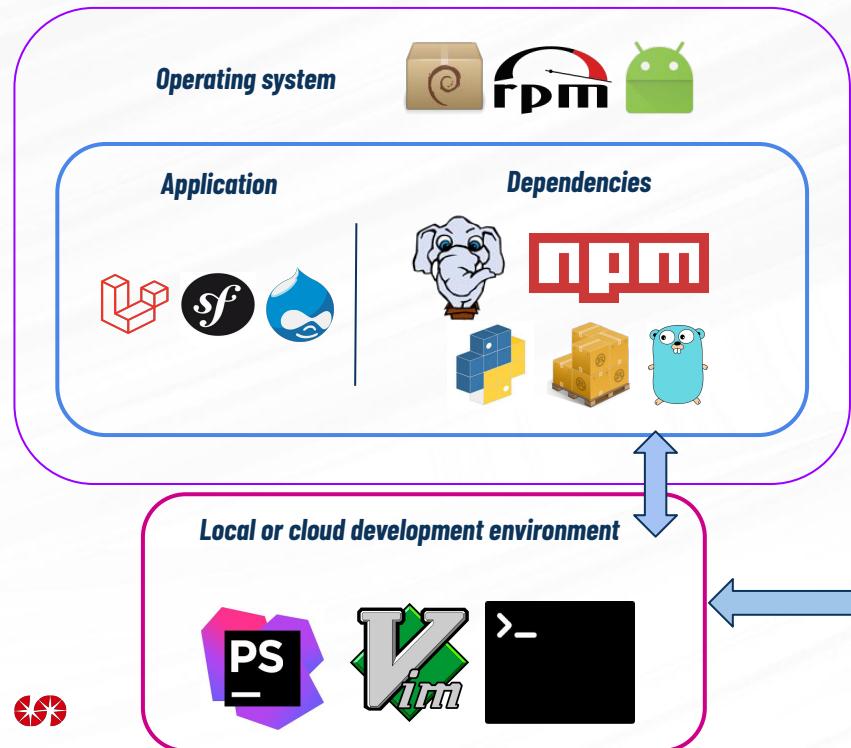
A MODERN PHP APPLICATION



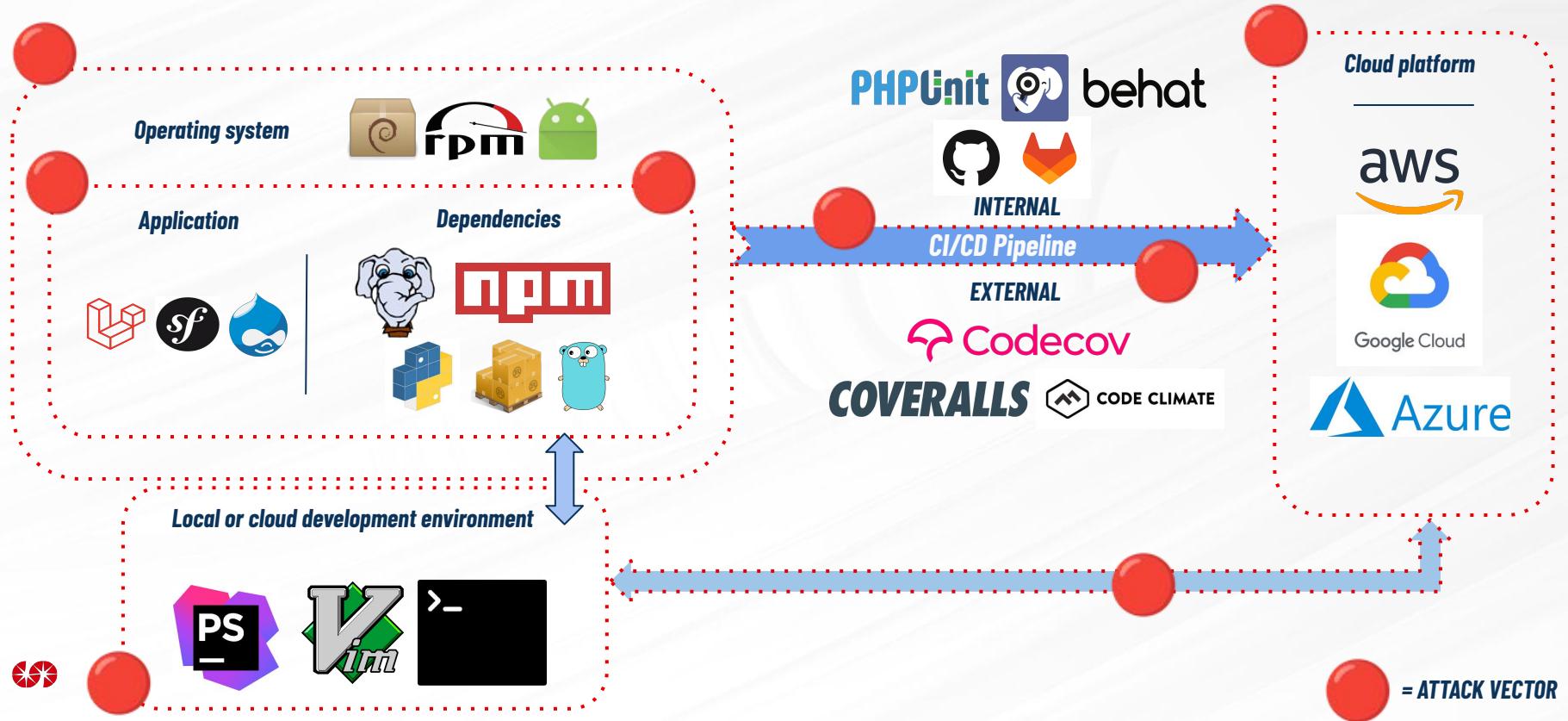
A MODERN PHP APPLICATION

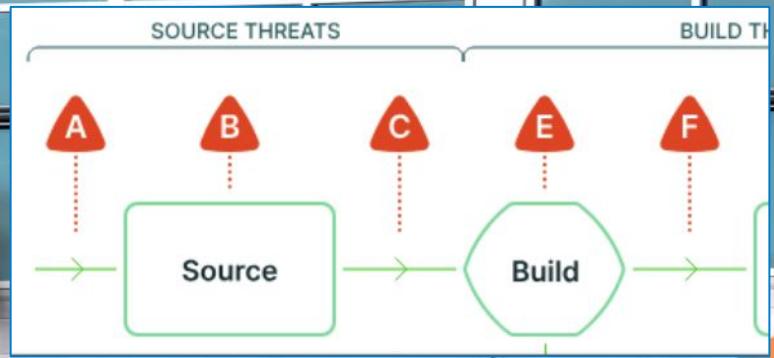


A MODERN PHP APPLICATION



A MODERN PHP APPLICATION





solarwinds



2020

About 18,000 customers of SolarWinds installed the infected updates, including firms like Microsoft (Cisco, Intel, Deloitte) and top government US agencies like Pentagon, Homeland security, National Nuclear Security etc.

April 2024

SSH BACKDOOR VIA LIBLZMA

The story of the XZ failed (*by chance*) attack



<https://mastodon.social/@AndresFreundTec/112180083704606941>



AndresFreundTec

@AndresFreundTec@mastodon.social

I was doing some micro-benchmarking at the time, needed to quiesce the system to reduce noise. Saw sshd processes were using a surprising amount of CPU, despite immediately failing because of wrong usernames etc. Profiled sshd, showing lots of cpu time in liblzma, with perf unable to attribute it to a symbol. Got suspicious. Recalled that I had seen an odd valgrind complaint in automated testing of postgres, a few weeks earlier, after package updates.

Really required a lot of coincidences.

Mar 29, 2024, 07:32 PM · · Web





Products

Services

Publications

Resources

What's new

Follow @Openwall on Twitter for new release announcements and other news

[<prev] [next>] [thread-next>] [day] [month] [year] [list]

Date: Fri, 29 Mar 2024 08:51:26 -0700

From: Andres Freund <andres@...razel.de>

To: oss-security@...ts.openwall.com

Subject: backdoor in upstream xz/liblzma leading to ssh server compromise

Hi,

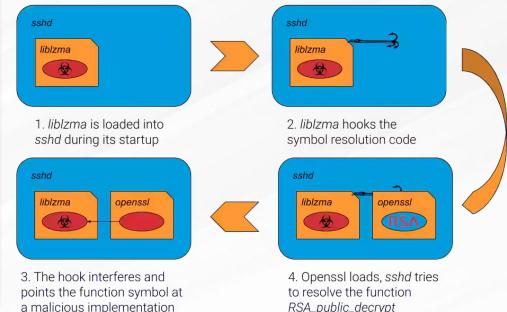
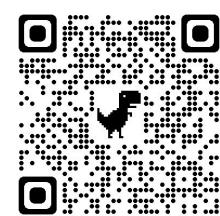
After observing a few odd symptoms around liblzma (part of the xz package) on Debian sid installations over the last weeks (logins with ssh taking a lot of CPU, valgrind errors) I figured out the answer:

The upstream xz repository and the xz tarballs have been backdoored.

At first I thought this was a compromise of debian's package, but it turns out to be upstream.

== Compromised Release Tarball ==

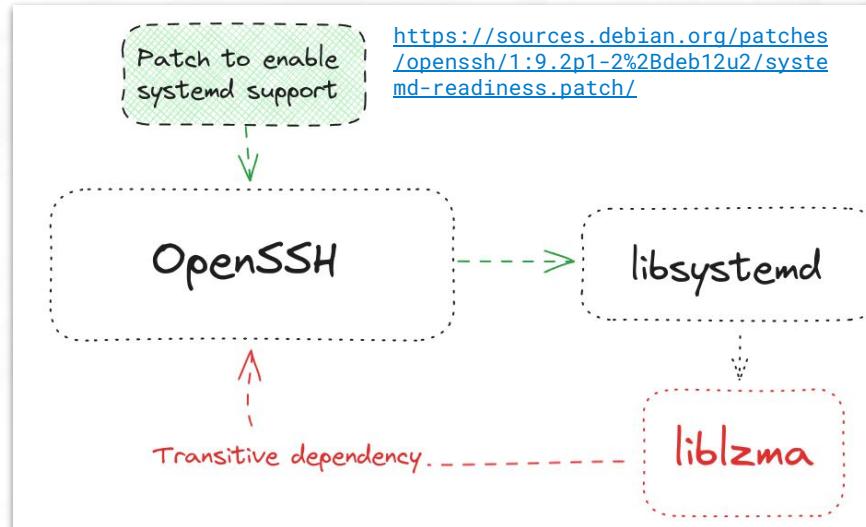
One portion of the backdoor is *solely in the distributed tarballs*. For easier reference, here's a link to debian's import of the tarball, but it is also present in the tarballs for 5.6.0 and 5.6.1:



What is XZ and why SSH has been impacted ?

XZ Utils and its underlying library, **liblzma**, are **open-source projects that implement LZMA compression** and decompression. They are **included in many Linux distributions** out of the box.

OpenSSH does not depend on XZ, however Debian and several other distributions - patch it to support systemd notification, and eventually, **libsystemd does depend on liblzma**.



XZ attack timeline



Github Activity Summary (user: JiaT75)

Repository:
<https://github.com/tukaani-project/xz>



2021

User Jia Tan (JiaT75)
creates his Github Account

JiaT75's first commit
to the XZ repo

2022-02-06

PR opened in oss-fuzz to
disable ifunc for fuzzing
builds. Allegedly to mask the
malicious changes.

2023-07-08

Obfuscated/encrypted stages binary backdoor
hidden in two test files:

- [tests/files/bad-3-corrupt_lzma2.xz](#)
- [tests/files/good-large_compressed.lzma](#)

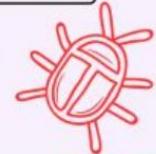
2024-03-09

2024-02-16

2023-06-28

Potential infrastructure testing:
liblzma: "Add ifunc implementation
to crc64_fast.c."

Malicious "build-to-host.m4" file added
to .gitignore, later incorporated to the
package release.



xz/libzma
v5.6.0 & v5.6.1

Packaged in the final releases



Source: https://www.linkedin.com/posts/thomas-roccia_infosec-xz-cybersecurity-activity-7180110597139697664-nzJL



Open source has won



It won yes, but is it still sustainable ?

Re: [xz-devel] XZ for Java

Lasse Collin | Wed, 08 Jun 2022 03:28:08 -0700

On 2022-06-07 Jigar Kumar wrote:

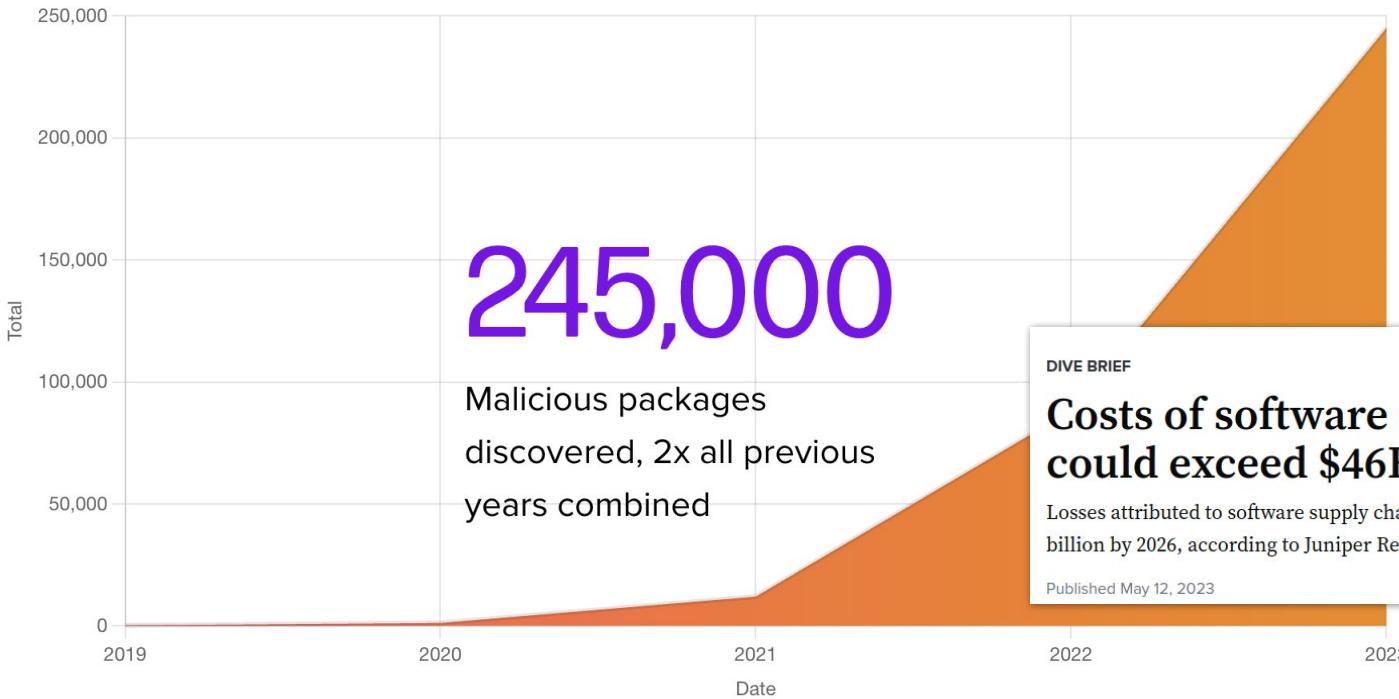
> Progress will not happen until there is new maintainer. XZ for C has
> sparse commit log too. Dennis you are better off waiting until new
> maintainer happens or fork yourself. Submitting patches here has no
> purpose these days. The current maintainer lost interest or doesn't
> care to maintain anymore. It is sad to see for a repo like this.

I haven't lost interest but my ability to care has been fairly limited mostly due to longterm mental health issues but also due to some other things. Recently I've worked off-list a bit with Jia Tan on XZ Utils and perhaps he will have a bigger role in the future, we'll see.



It's also good to keep in mind that this is an unpaid hobby project.

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



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

NATIONAL CYBERSECURITY STRATEGY

MARCH 2023



<https://linuxfoundation.eu/cyber-resilience-act>



The graphic features the European Commission logo (blue square with yellow stars) next to stylized grey wavy lines. To the right is a large black 'CE' mark. Below the logo, the text 'European Commission' is written. At the bottom left, there's a circular image showing hands interacting with a futuristic interface featuring padlock icons. The text 'CYBER RESILIENCE ACT' is prominently displayed in white, followed by the subtitle 'New EU cybersecurity rules ensure more secure hardware and software products'. Hashtags '#DigitalEU #SecurityUnion #Cybersecurity' are listed at the bottom. The date 'SEPTEMBER 2022 – UPDATED DECEMBER 2023' is at the very bottom right.

CYBER RESILIENCE ACT

New EU cybersecurity rules ensure more secure hardware and software products

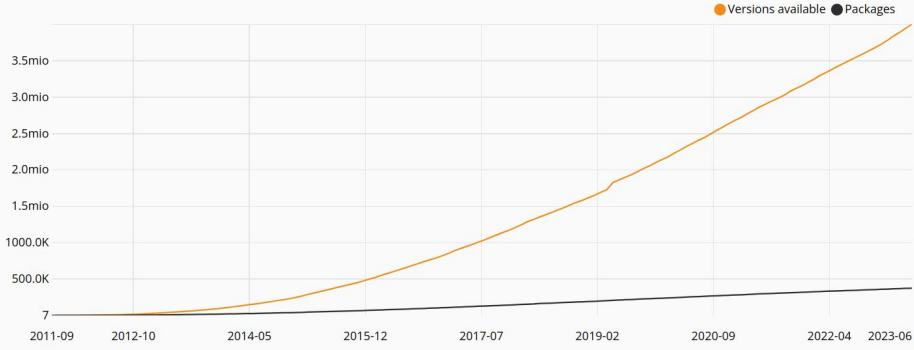
#DigitalEU #SecurityUnion #Cybersecurity

SEPTEMBER 2022 – UPDATED DECEMBER 2023

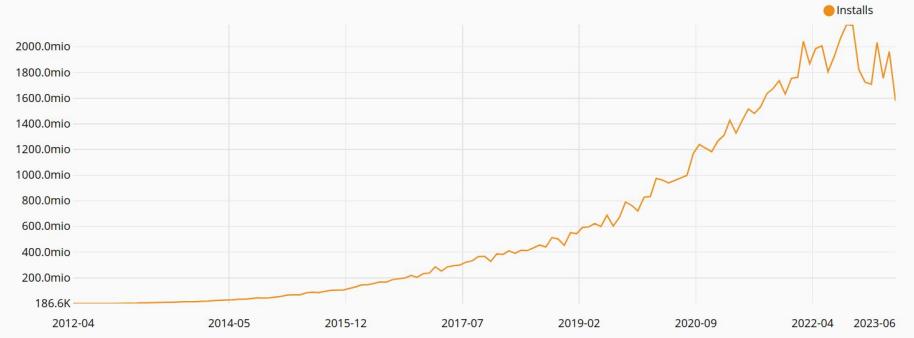
STATE OF THE PHP ECOSYSTEM

COMPOSER: THE PHP PACKAGE MANAGER

Packages/versions over time



Packages installed per month



- Invented in 2012 by **Nils Adermann** and **Jordi Boggiano**.
- **Standard de-facto** for the PHP package management.
- packagist.org hosts more than 300k packages and 2.5M revisions.

COMPOSER BUILT-IN SECURITY PROTECTIONS



<https://blog.packagist.com/preventing-dependency-hijacking>



THE LATEST SUPPLY CHAIN ATTACKS ON PHP

April 29, 2021

[PHP Supply Chain Attack on Composer](#)

"A critical vulnerability in the source code of Composer which is used by Packagist. It allowed us to execute arbitrary system commands on the Packagist.org server"

October 4, 2022

[Securing Developer Tools: A New Supply Chain Attack on PHP](#)

"A new critical vulnerability in similar components. It allowed taking control of the server distributing information about existing PHP software packages, and ultimately compromising every organization that uses them"

March 29, 2022

[PHP Supply Chain Attack on PEAR](#)

"In this article we present two bugs, both exploitable for more than 15 years. An attacker exploiting the first one could take over any developer account and publish malicious releases, while the second bug would allow the attacker to gain persistent access to the central PEAR server."

May 3, 2023

[Packagist.org maintainer account takeover](#)

"An attacker accessed an inactive account on Packagist.org for a period of time but still had access to a total of 14 packages. The attacker forked each of the packages and replaced the package description in composer.json with their own message but did not otherwise make any malicious changes"

And counting

<https://www.sonatype.com/resources/vulnerability-timeline>



SO WHAT ? WHERE SHOULD WE START ?



INTEGRITY, TRUST AND DEPENDENCIES

What is the ***trusting model*** between
us and ***digital artifacts?***

How can i be sure that ***what I'm running***
is coming from a ***trusted source?***



Reflections on Trusting Trust

To what extent should one trust a statement that a program is free of Trojan horses? Perhaps it is more important to trust the people who wrote the software.

MORAL

The moral is obvious. You can't trust code that you did not totally create yourself. (Especially code from companies that employ people like me.) No amount of source-level verification or scrutiny will protect you from using untrusted code. I

KEN THOMPSON

SECURE SOFTWARE SUPPLY CHAIN CHECKLIST

- ✓ Who built it, when and how
(Signatures and Provenance Attestations)
- ✓ The list of things who made the artifact
(SBOM - Software Bill of Material)

DIGITAL SIGNATURES 101

Integrity

Ensure the data signed was not altered.

Authenticity

Attest that the data was sent by the signer.

Non-repudiation

Ensure that the signer cannot deny the authenticity of the signature.



Managing keys is *hard*

Distribution, Storage, Compromise



DIGITAL SIGNATURES - SIGSTORE

Sigstore is an OSS project under the umbrella of [OpenSSF](#) foundation.

Fast growing community and mainstream adopted

Used in **Kubernetes** and many other big vendors

(Github, Rubygems, Arch Linux etc..)

<https://openssf.org/community/sigstore>



In collaboration with



DIGITAL SIGNATURES - SIGSTORE

Keyless signing of any software artifact

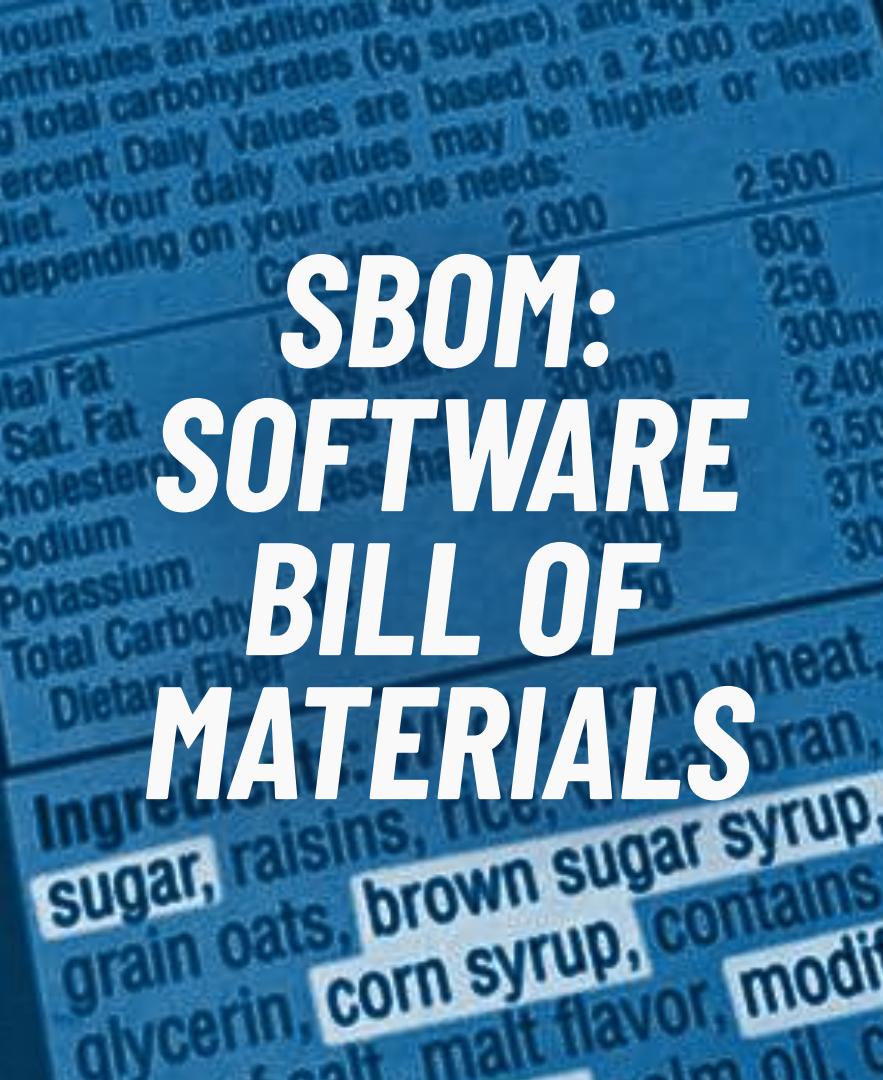
Signatures metadata are stored in a [public tamper-resistant log](#)

Signatures are stored alongside images in *OCI registry*



In collaboration with





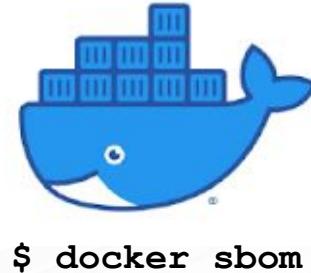
SBOM: SOFTWARE BILL OF MATERIALS

**A list of “ingredients”
for a software artifact**

Can be used for:

- Vulnerability scanning
- Software transparency
- License policy
- Find abandoned dependencies

SBOM - Tools



<https://github.com/CycloneDX/cyclonedx-php-composer>



KNOW YOUR DEPENDENCIES



OSV

<https://osv.dev>

OSV

Vulnerability Database

Blog

FAQ

A distributed vulnerability database for Open Source

An open, precise, and distributed approach to producing and consuming vulnerability information for open source.

[Search Vulnerability Database](#)

[Use the API](#)

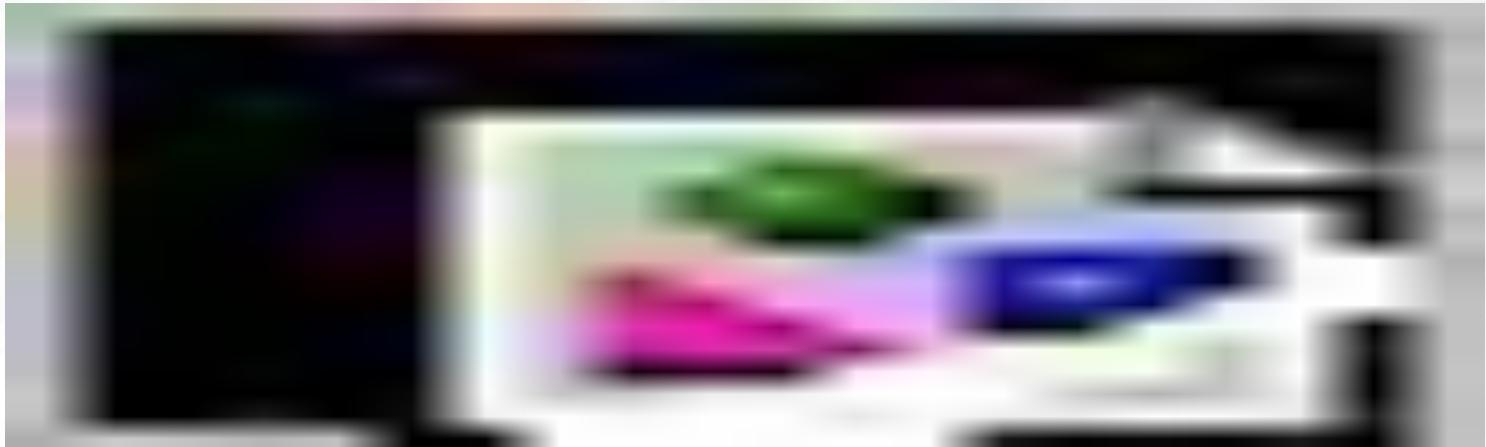
[CLI Tools](#)

<https://ossf.github.io/osv-schema>

<https://github.com/google/osv-scanner>

OpenSSF Scorecard

<https://scorecard.dev>



Checks: Code vulnerabilities, Maintenance, Continuous testing, Source risk assessment, Build risk assessment



Automated dependencies management

<https://github.com/renovatebot/renovate> - <https://github.com/dependabot>



Dependency Dashboard #6

Open 6 tasks renovate bot opened this issue 6 days ago · 0 comments

[Edit](#) [New issue](#)

[Write](#) [Preview](#)

This issue lists Renovate updates and detected dependencies. Read the [Dependency Dashboard](#) docs to learn more.

You can access the [Renovate App Dashboard](#) to see logs for the Renovate jobs on your repository.

Open

These updates have all been created already. Click a checkbox below to force a retry/rebase of any.

Update dependency lodash to v4.17.21
 Update date-io monorepo to v2.14.0 (`@date-io/date-fns`, `@date-io/moment`)
 Update dependency commander to v9
 Click on this checkbox to rebase all open PRs at once

Ignored or Blocked

These are blocked by an existing closed PR and will not be recreated unless you click a checkbox below.

Update dependency php to v8.1

Detected dependencies

▶ dockerfile
▶ github-actions

Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

Milestone
No milestone

Development
Create a branch for this issue or link a pull request.

Notifications
Customize [Subscribe](#)
You're not receiving notifications from this thread.

0 participants

[Lock conversation](#)



Takeaways

- Digital Signatures with Sigstore and Software Bills of materials.
- More informed choice of external dependencies with [OSV](#),
[OpenSSF Scorecard](#) and [deps.dev](#)
- Automate your dependencies management with Github
DependaBot or Renovate for all other platforms.
- Demo of Sigstore, Syft, Grype, Chainguard images:
https://www.youtube.com/watch?v=8osHp_h9bYU

Demo



THANKS

