CSC405

Computer Security

Software Supply Chain Security

Acknowledgements: Laurie Williams, Yasemin Acar, Michel Cukier, William Enck, Alexandros Kapravelos, Christian Kästner, Dominik Wermke

Learning Objectives

1. Explain the structure of software supply chains
2. Outline common risks associated with code dependencies, including dependency confusion
3. Outline good security practices for build infrastructure, including signing and reproducible builds.
4. Describe key challenges for securing the software supply chain in the context of humans.

Overview for Today

1. Intro: (Software) Supply Chain
2. SSC Areas:
    a. Code Dependencies
    b. Build Infrastructure
    c. The Human Factor

Supply Chain

Complex logistics system

- Process of producing and delivering a product or service
- Network of entities like suppliers, manufacturers, distributors, and retailers

Supply Chain Problems

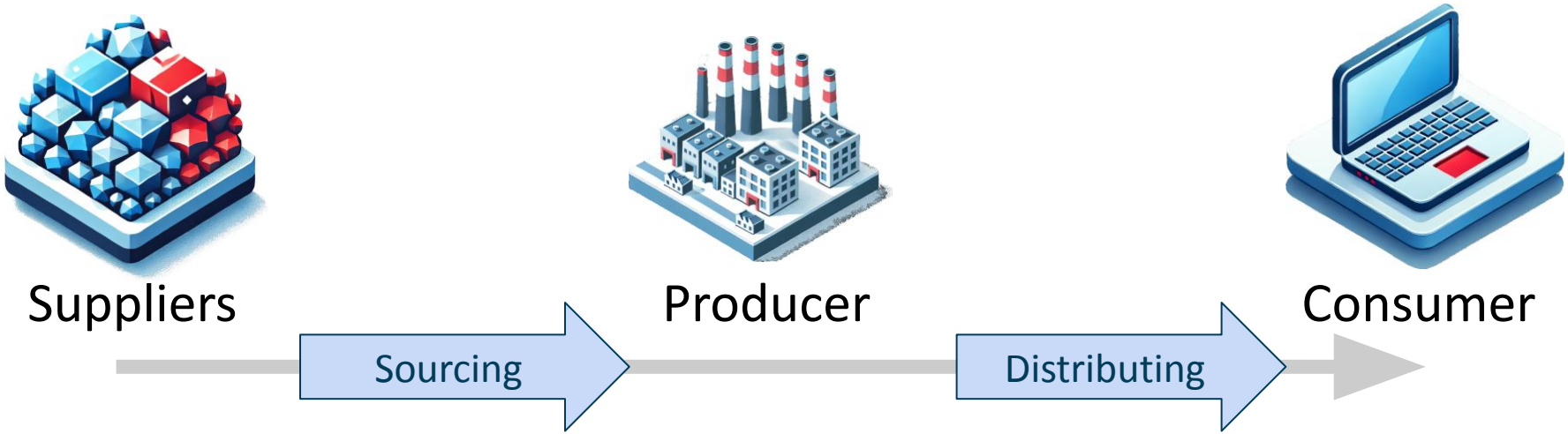
Supply Chain Problems 2

Port Strike Could Reignite Toilet Paper Shortage — But Don’t Panic, Expert Says

October 2, 2024 | Andrew Moore | 2-min. read

Supply Chain

Rough outline of an industry supply chain:



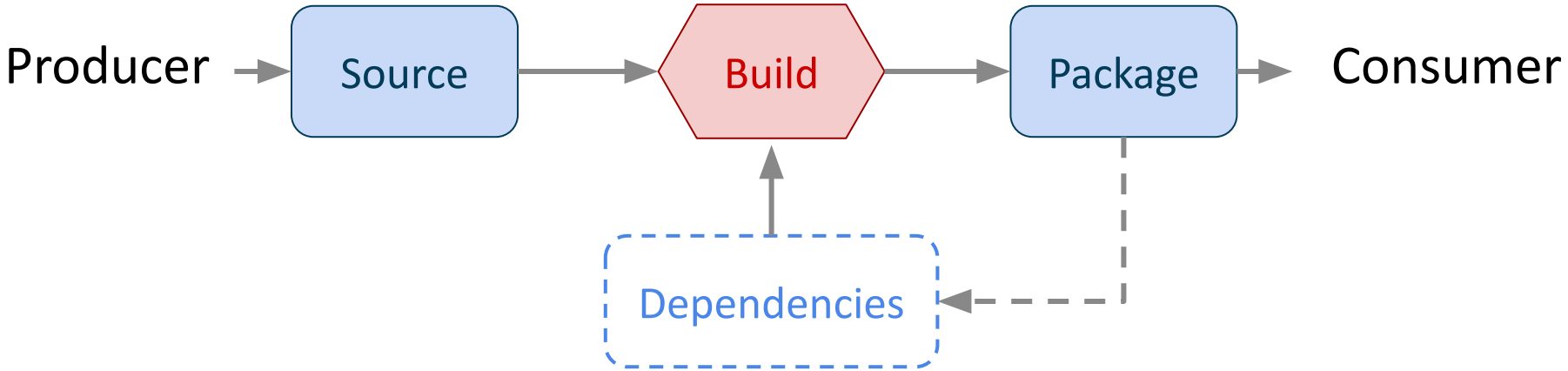
Software Supply Chain?

What would you consider part of a software supply chain?

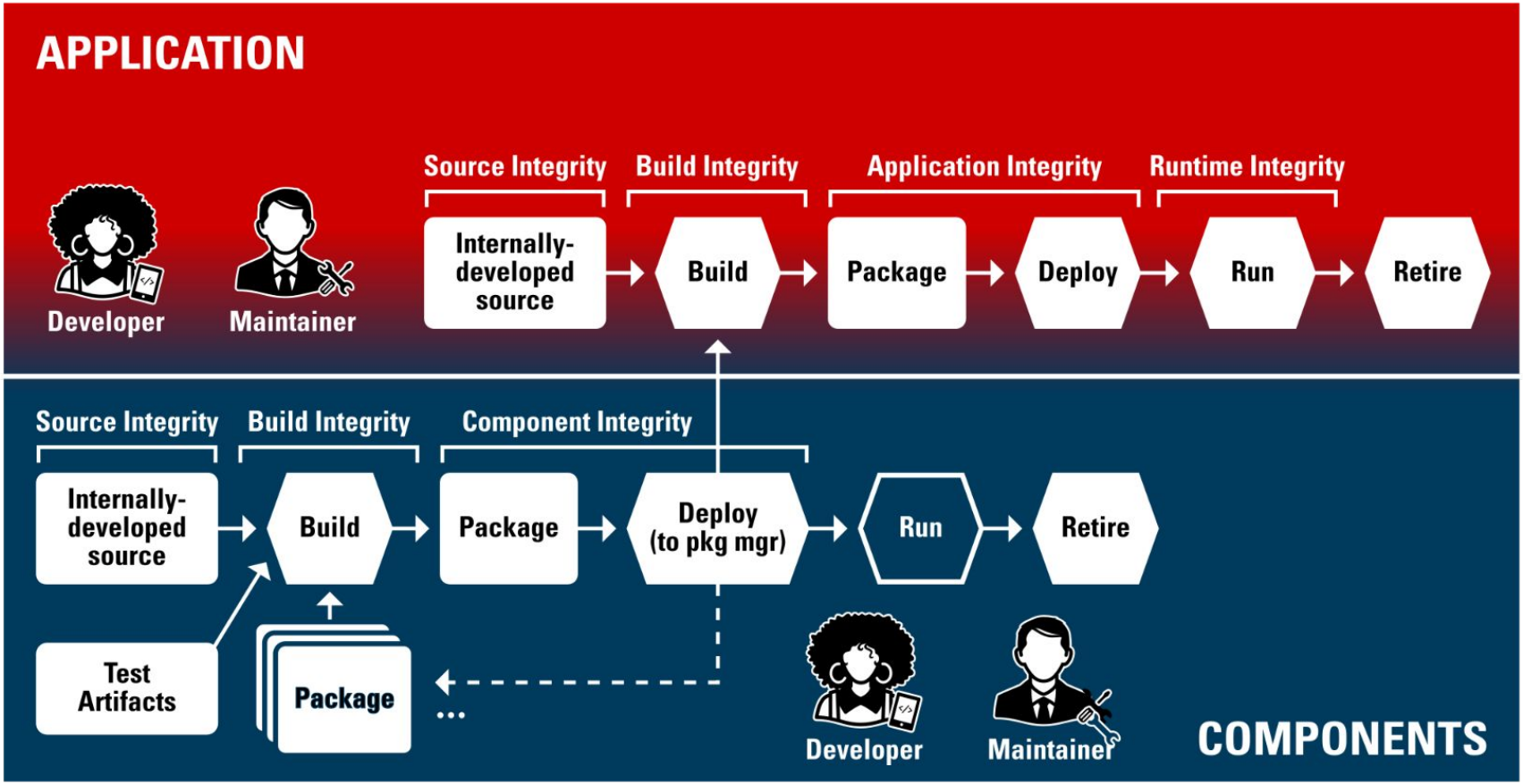
Software Supply Chain

- Source code
- Dependencies
- Operational infrastructure
- Dev tools
- CI/CD tools and pipelines
- Artifact repositories
- Distribution systems
- …

Software Supply Chain



Application vs. Components




THE SUPPLY CHAIN
IS UNDER
ATTACK!

TechTarget

Microsoft: Nation-state threats, zero-day attacks increasing

"While zero-day vulnerability attacks tend to initially target a ... the software supply chain to exploiting the IT services supply chain,...

ZDNET

Supply chain hacks are on the rise. But most companies aren't prepared

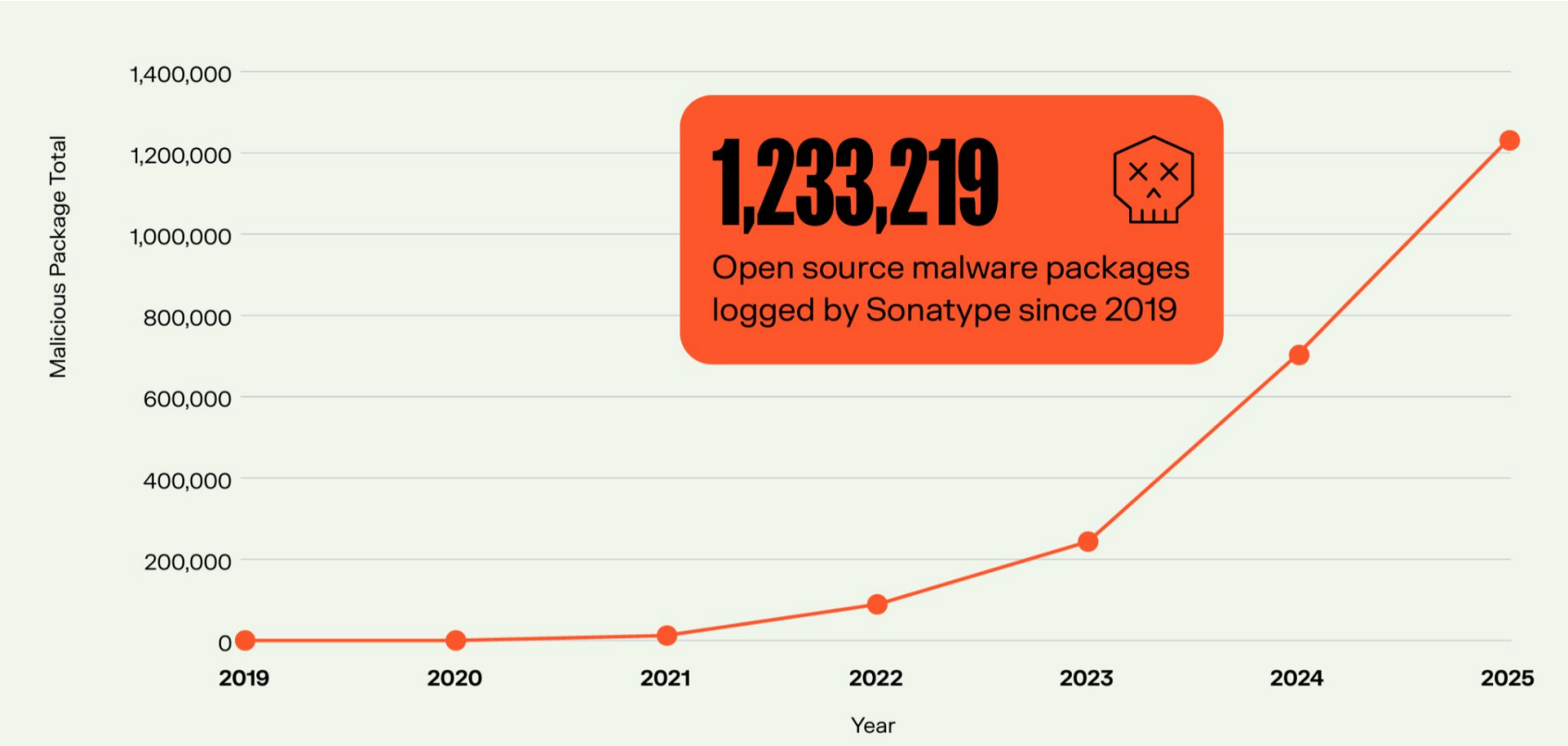
"Supply chain attacks are a major cyber threat facing ... US spy agency, the NSA, last month published its software supply chain guidance,...

TechTarget

NPM malware attack goes unnoticed for a year

Software supply chain attacks have become a growing trend in recent years as criminals have discovered that by infecting the code dependencies...

Rise in SSC Attacks



https://www.sonatype.com/state-of-the-software-supply-chain/Introduction

Moments in SSC Security

December 2020

Build infrastructure

November 2021

LOG4J

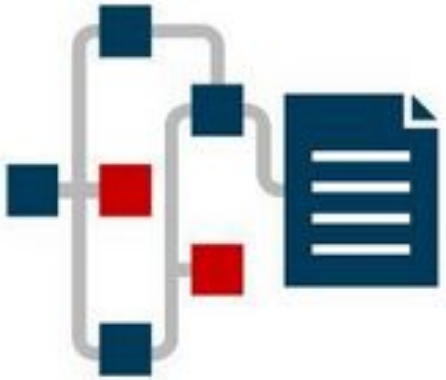
Vulnerable components

March 2024

Malicious components, Build infrastructure, Humans

Presidential Order

President's Executive Order 14028 - Improving the Nation's Cybersecurity

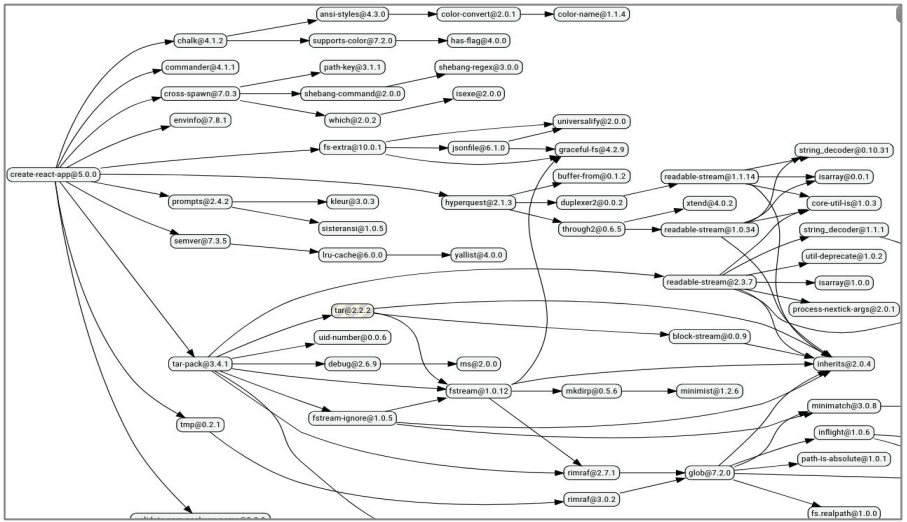


Code Dependencies

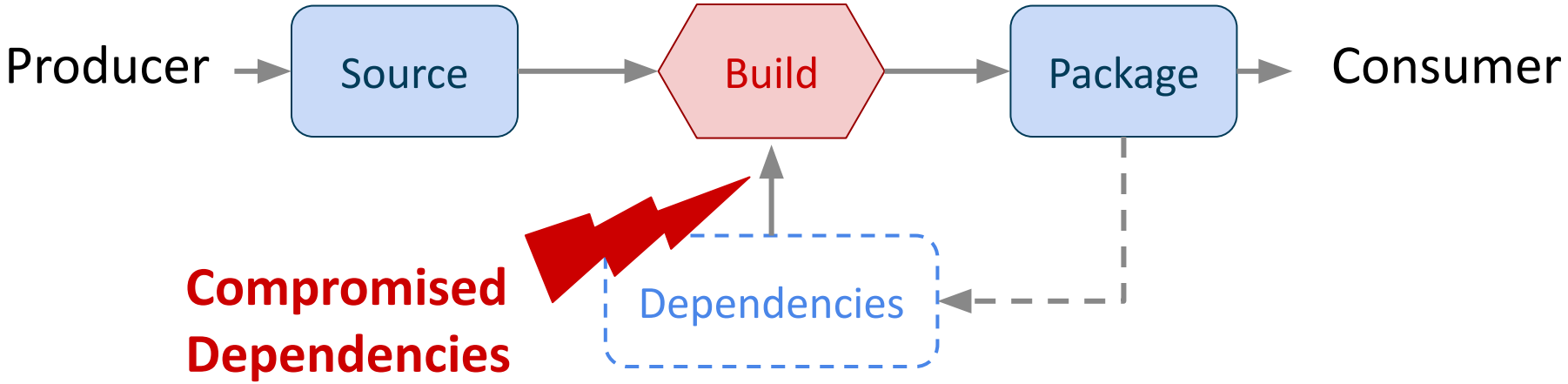
Code Dependencies

External libraries, frameworks, or modules in a software project

Often downloaded and updated using a package manager (npm, pip, maven) from package repositories (npm registry, PyPI, Maven Central)



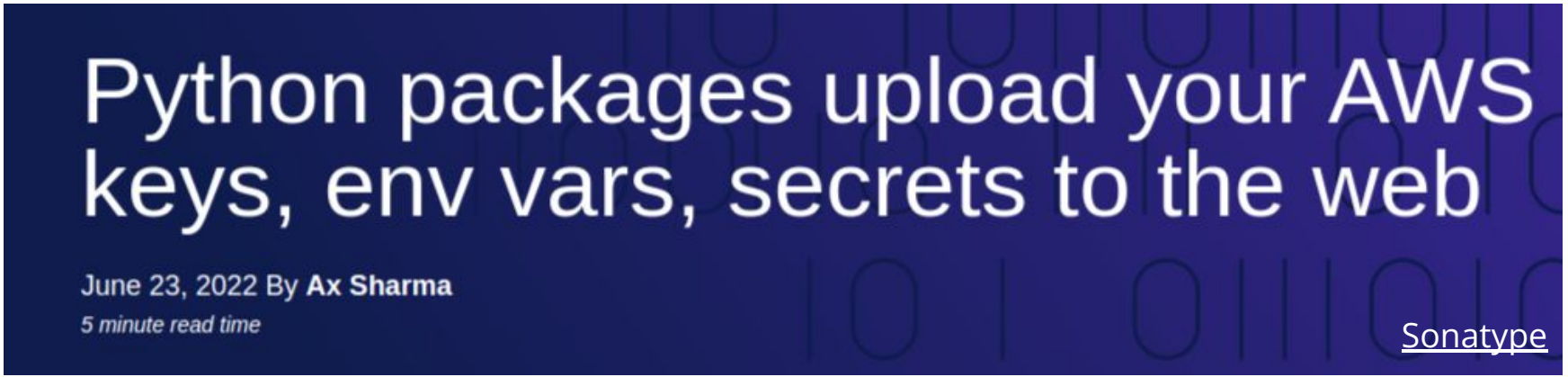
Dependency Attacks



Attack: Malicious Package

Malicious package from scratch

- Actual development + advertising, can turn malicious later
- loglib-modules (targeting developers familiar with the legitimate 'loglib' library)



Ladisa et al. "Sok: Taxonomy of attacks on open-source software supply chains." IEEE S&P, 2023.

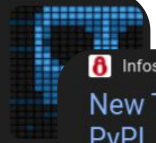
Name Confusion Attacks

Infosecurity Magazine

Researchers Uncover 700+ Malicious Open Source Packages

Security researchers have discovered another sizeable haul of malicious packages on the npm and PyPI open source registries, which could cause issues if...

Feb 13, 2023



Infosecurity Magazine

New Typosquatting and Repojacking Tactics Uncovered on PyPI

The recent discovery of NP6HelperHttptest and NP6HelperHttper on PyPI exemplifies such tactics, exploiting similarities with legitimate NP6...

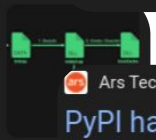
Feb 20, 2024

The Hacker News

Lazarus Exploits Typos to Sneak PyPI Malware into Dev Systems

North Korean hackers infiltrated PyPI with malware-laden packages, exploiting common typos.

Feb 29, 2024



Ars Technica

PyPI halted new users and projects while it fended off supply-chain attack

Automation is making attacks on open source code repositories harder to fight.

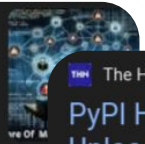
1 month ago

GBHackers on Security

New Malicious PyPI Packages Use DLL Sideloading In A Supply Chain Attack

Researchers have discovered that threat actors have been using open-source platforms and codes for several purposes, such as hosting C2 infrastructure,...

Feb 21, 2024



The Hacker News

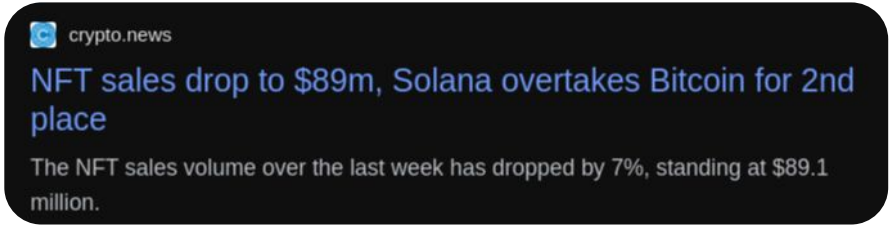
PyPI Halts Sign-Ups Amid Surge of Malicious Package Uploads Targeting Developers

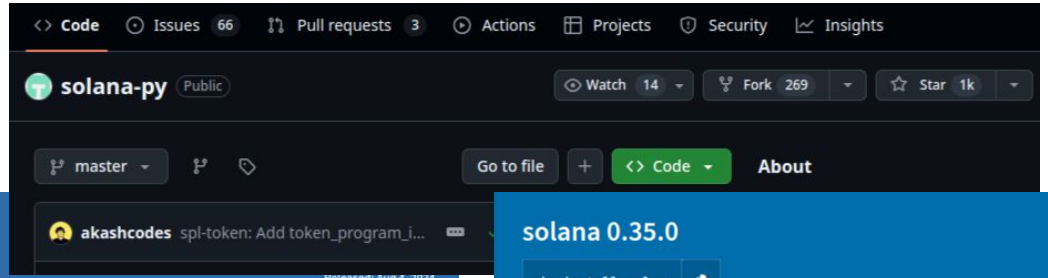
PyPI temporarily shut down new user sign-ups and project creation to combat a malicious malware upload campaign.

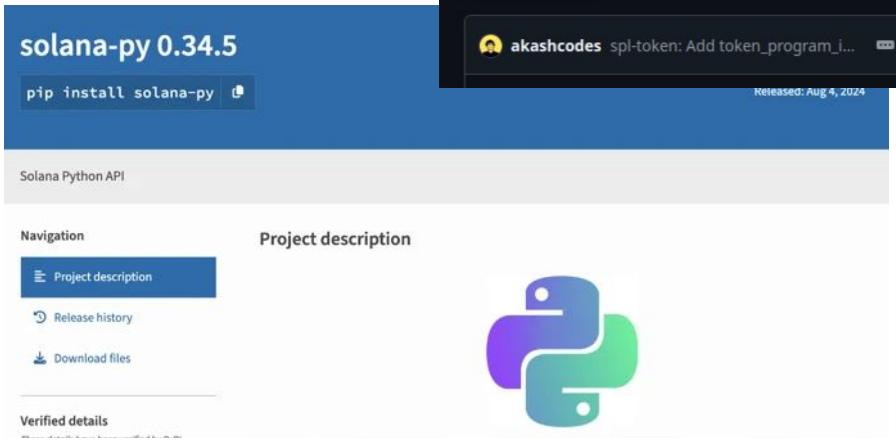
1 month ago

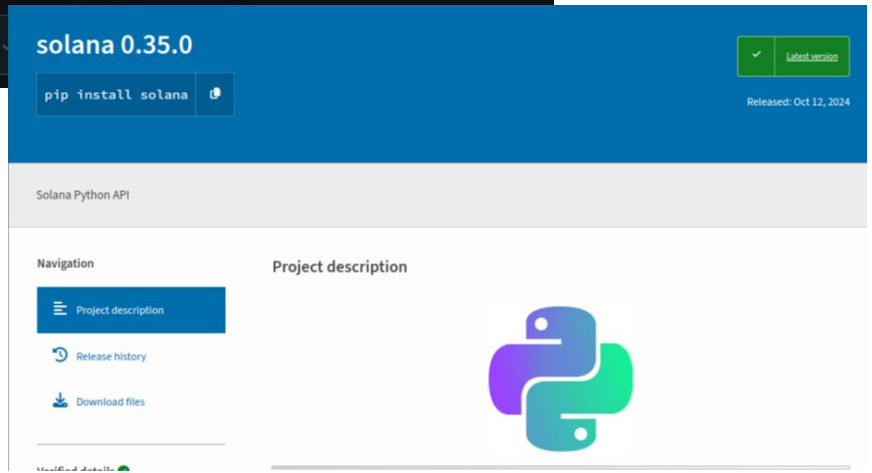
solana-py

Which is the real one?







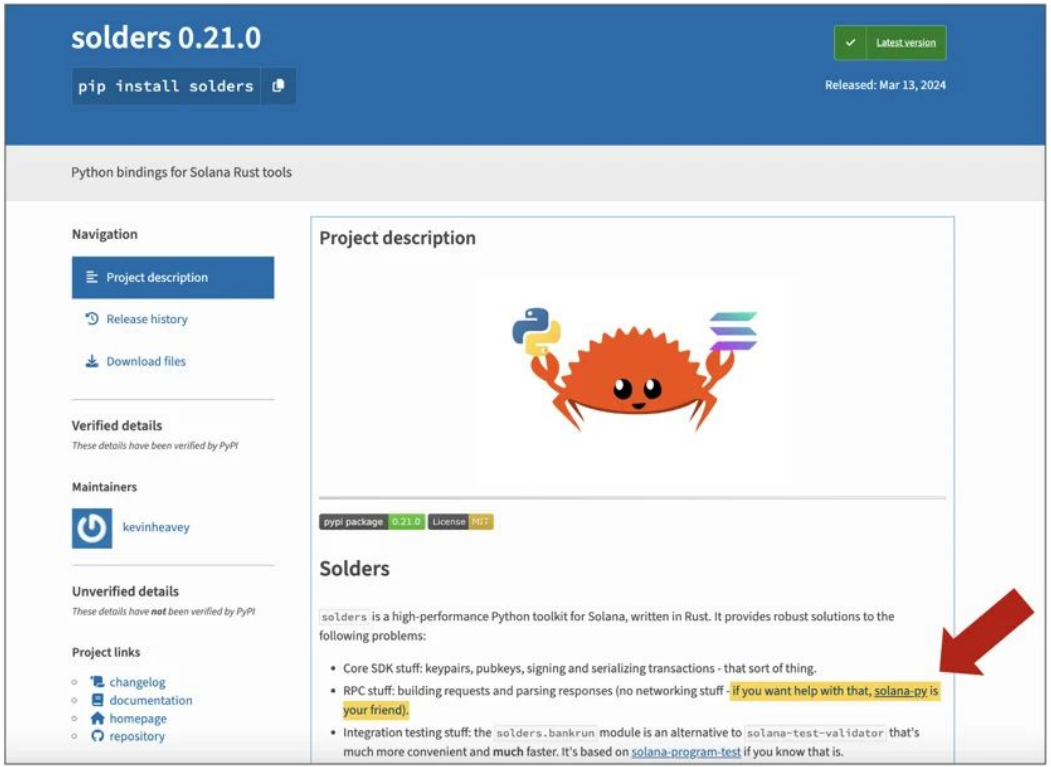


https://www.sonatype.com/blog/an-ideal-pypi-typosquat-solana-py-is-here-to-steal-your-crypto-keys

Even More Confusing

Other legitimate projects (wrongly) refer to solana-py!

Malicious package steals wallet keys



https://www.sonatype.com/blog/an-ideal-pypi-typosquat-solana-py-is-here-to-steal-your-crypto-keys

Attacks: Name Confusion

- Typosquatting (request vs. req7est)
- Combosquatting (openai-dev, python-4)
- Separators (python-3 vs. python_3)
- Scope confusion (@npm/test vs npm-test)
- Shadow built-in package (subprocess)
  - Also for common strings in error messages (TypeError) because people might just run install
- Brandjacking (aws-official-openai)
  - Works also for scopes (awsreal/openai)
- Shadowing / Dependency Confusion (company-internal-foo)
  - Trick internal proxies to pull from public repo

Dependency Confusion

Paypal's leaked package.json

Public packages

Not on npm (internal packages?)

```
"dependencies": {
  "express": "^4.3.0",
  "dustjs-helpers": "~1.6.3",
  "continuation-local-storage": "^3.1.0",
  "pplogger": "^0.2",
  "auth-paypal": "^2.0.0",
  "wurfl-paypal": "^1.0.0",
  "analytics-paypal": "~1.0.0"
}
```

Dependency Confusion: How I Hacked Into Apple, Microsoft and Dozens of Other Companies

The Story of a Novel Supply Chain Attack

Alex Birsan · Follow

11 min read · Feb 9, 2021

Where are internal packages pulled from? Are same name packages on public npm preferred?

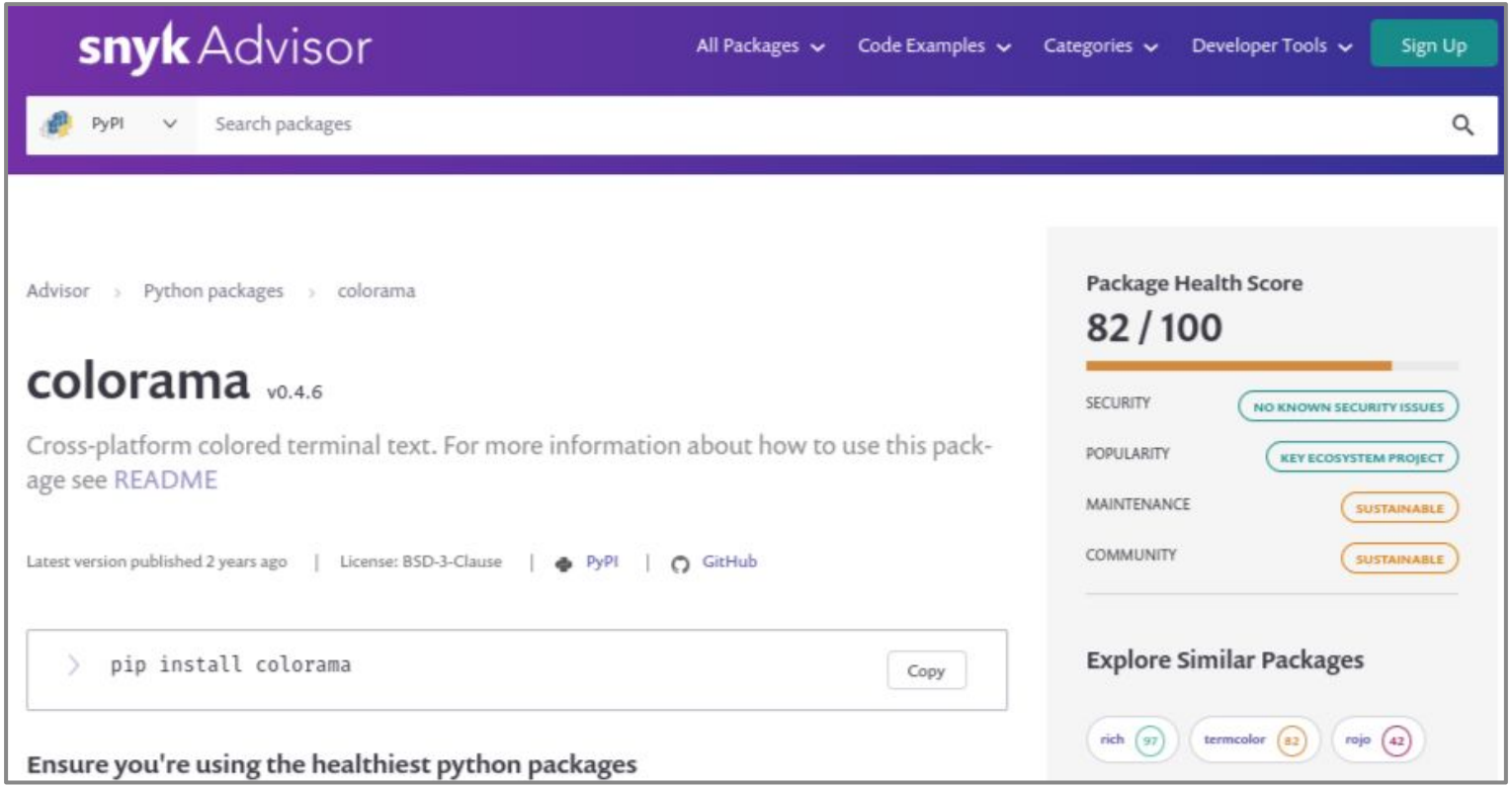
Attack: Register malicious packages with leaked (or guessed) internal names on npm (maybe with newer version number), wait for companies to pull malicious dependencies in.

https://medium.com/@alex.birsan/dependency-confusion-4a5d60fec610

Confusion Defenses?

How would you defend yourself and/or the supply chain against these attacks?

Defense: Dashboards



Attack: Infiltrate Malicious Code

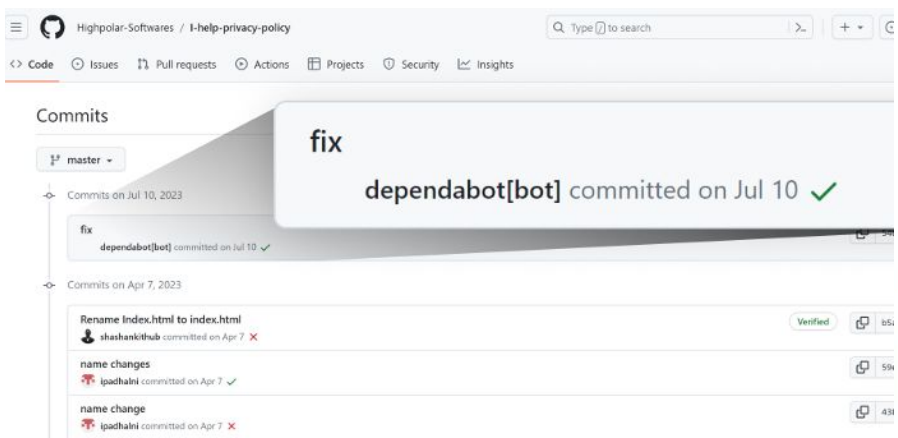
Infiltrate malicious code in existing package.

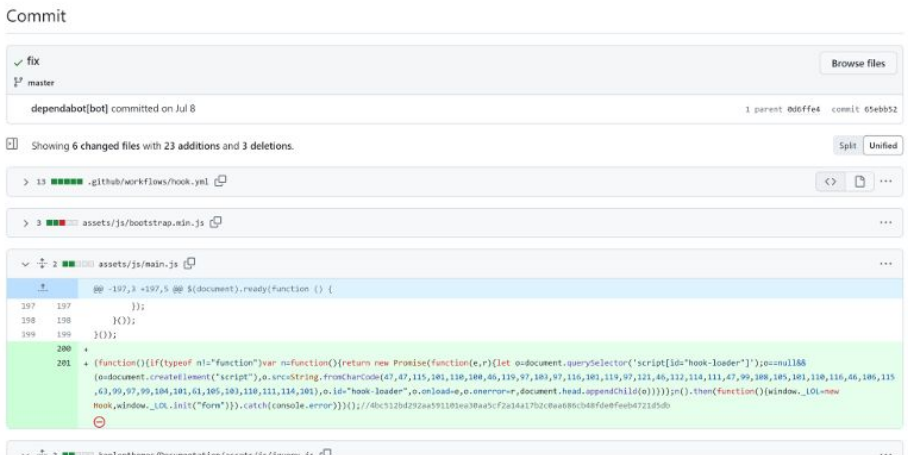
How could one accomplish this?

Malicious Pull Requests

Example: Fake Dependabot

- GitHub bot that automatically updates dependencies by opening pull requests
- Except when it is another account pretending to be Dependabot

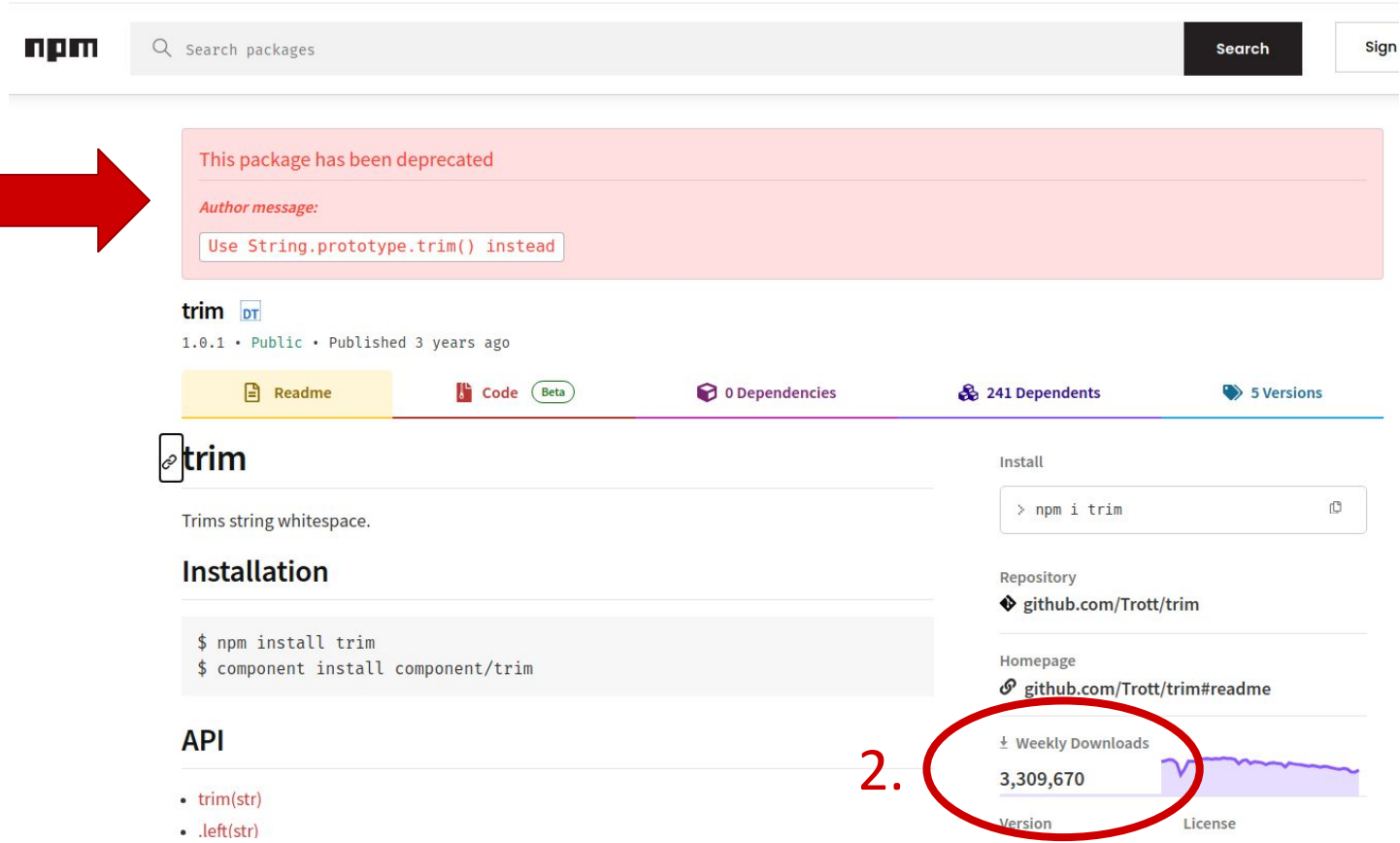




https://checkmarx.com/blog/surprise-when-dependabot-contributes-malicious-code/

Challenge: Abandoned Dependencies

1.



2.

Abandoned Dependencies

Ever looked at `npm install` output?

```
npm WARN deprecated babel-eslint@10.1.0: babel-eslint is now @babel/eslint-parser. This package will no longer receive
updates.
npm WARN deprecated chokidar@2.1.8: Chokidar 2 will break on node v14+. Upgrade to chokidar 3 with 15x less dependencie
s.
npm WARN deprecated svgo@1.3.2: This SVGO version is no longer supported. Upgrade to v2.x.x.
npm WARN deprecated querystring@0.2.1: The querystring API is considered Legacy. new code should use the URLSearchParam
s API instead.
npm WARN deprecated @hapi/joi@15.1.1: Switch to 'npm install joi'
npm WARN deprecated rollup-plugin-babel@4.4.0: This package has been deprecated and is no longer maintained. Please use
 @rollup/plugin-babel.
npm WARN deprecated fsevents@1.2.13: fsevents 1 will break on node v14+ and could be using insecure binaries. Upgrade t
o fsevents 2.
npm WARN deprecated uuid@3.4.0: Please upgrade  to version 7 or higher.  Older versions may use Math.random() in certai
n circumstances, which is known to be problematic.  See https://v8.dev/blog/math-random for details.
npm WARN deprecated querystring@0.2.0: The querystring API is considered Legacy. new code should use the URLSearchParam
s API instead.
npm WARN deprecated sane@4.1.0: some dependency vulnerabilities fixed, support for node < 10 dropped, and newer ECMAScr
ipt syntax/features added
npm WARN deprecated flatten@1.0.3: flatten is deprecated in favor of utility frameworks such as lodash.
npm WARN deprecated urix@0.1.0: Please see https://github.com/lydell/urix#deprecated
npm WARN deprecated @hapi/bourne@1.3.2: This version has been deprecated and is no longer supported or maintained
```

Abandoned Dependencies

Developers leave for all kinds of reasons

Abandoned project is not automatically broken, but no more fixes or new features (some companies enforce restrictions)

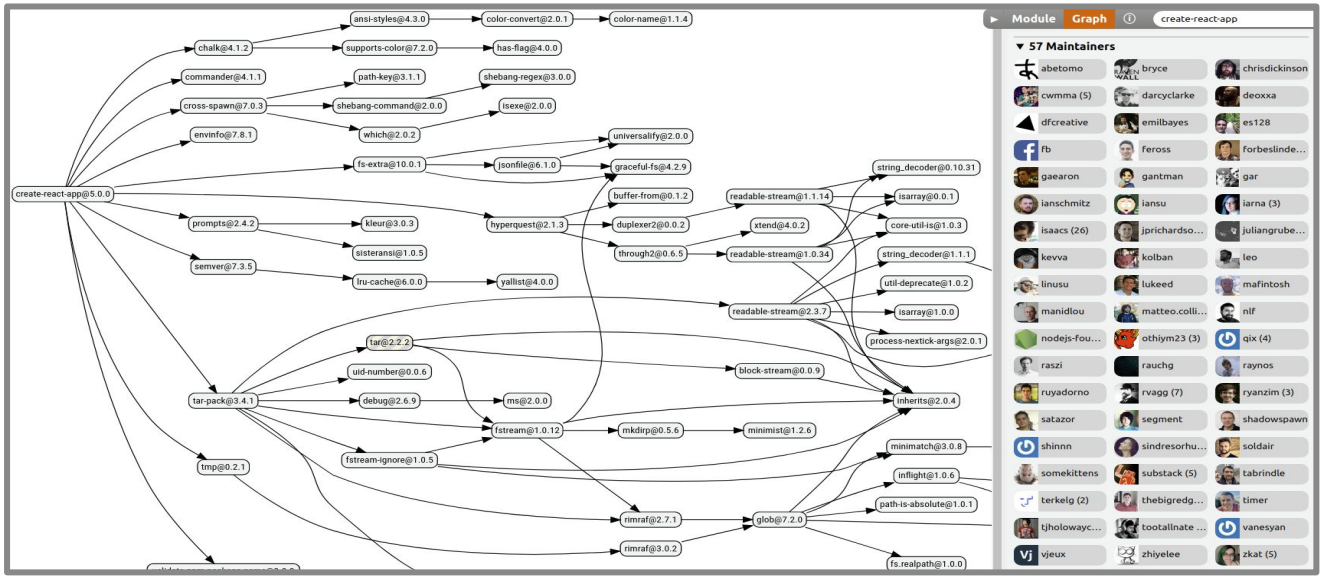
- missing features, security concerns
- time and effort for finding replacement

Q: Should we have expected free maintenance forever?

Q: How to avoid abandoned dependencies?

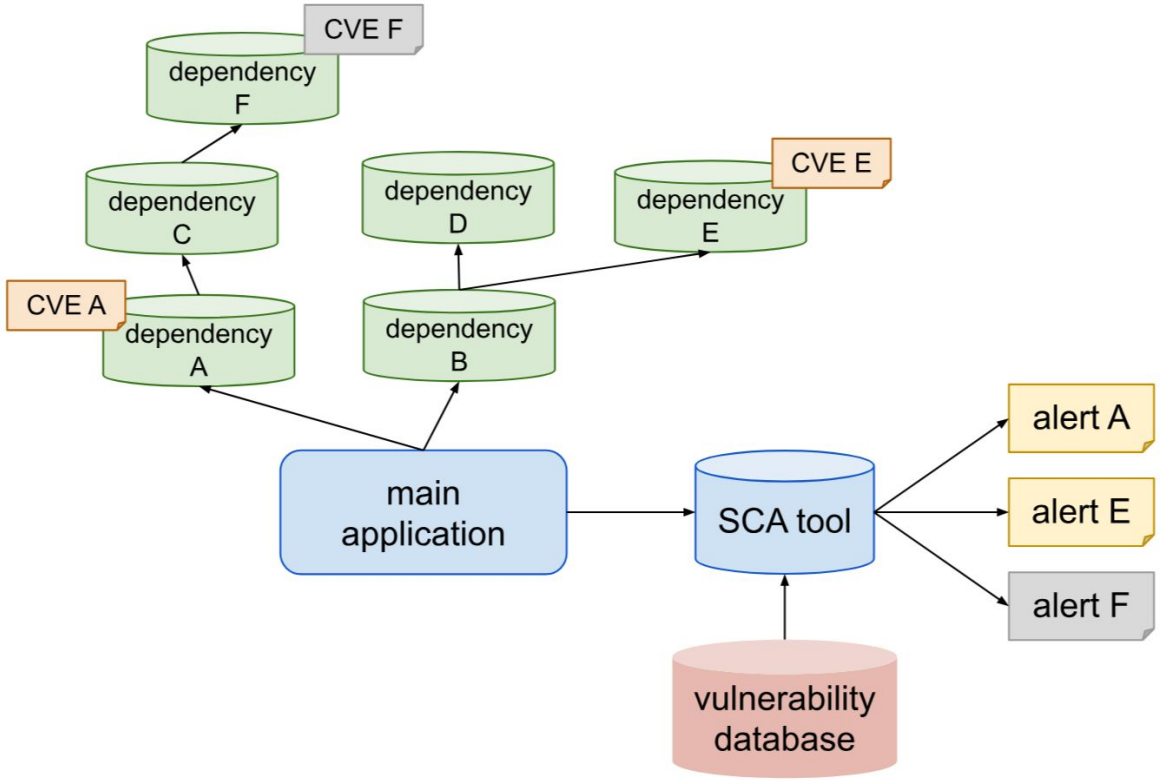
Problem: Transitive Dependencies

Software projects have 10s to 100s of direct and transitive dependencies

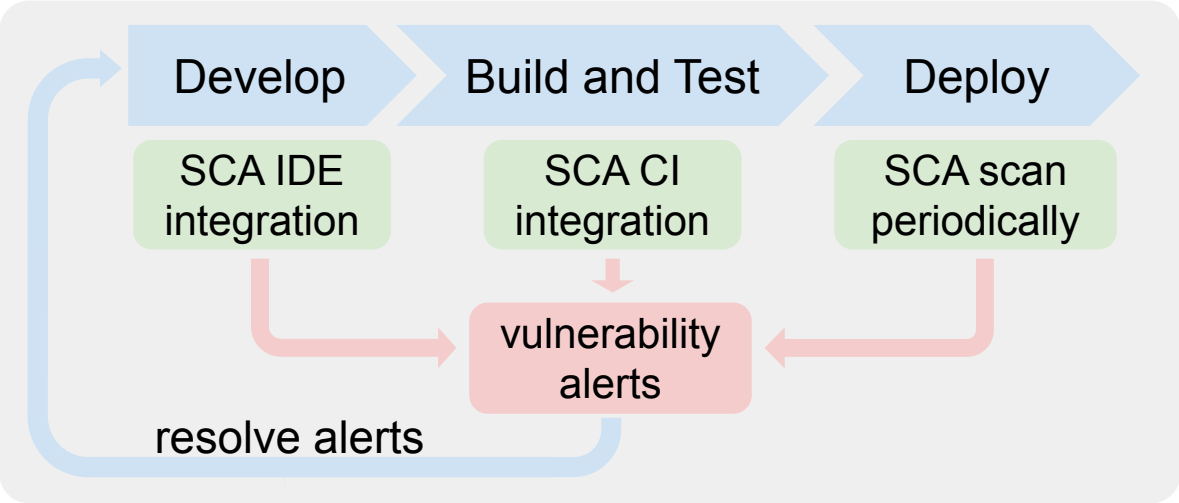


Software Composition Analysis

What is even in our application?

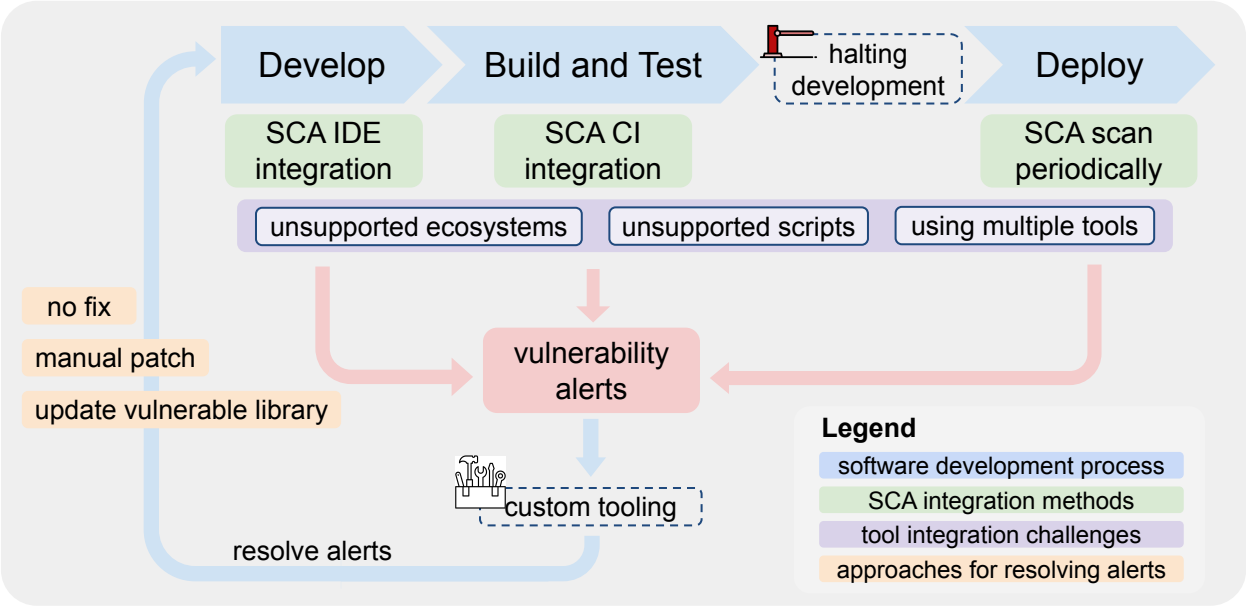


Ideal SCA Idea



SCA Reality

Many alerts, difficult to resolve without context



Recap: Presidential Order

President's Executive Order 14028 - Improving the Nation's Cybersecurity

NIST

- “(e) Within 90 days of publication of the preliminary guidelines [...] the Secretary of Commerce acting through the Director of NIST, in consultation with the heads of such agencies as the Director of NIST deems appropriate, shall issue guidance identifying practices that enhance the security of the software supply chain”
- (vii) providing a purchaser a Software Bill of Materials (SBOM) for each product directly or by publishing it on a public website;

Software Bill of Materials

(SBOM, inspired by supply chain BOMs)

- List of components used to build a software artifact
- Minimum elements (according to NIST):
  - data fields (baseline information about each software component)
  - automation support (the ability to generate SBOMs in machine- and human-readable formats)
  - practices and processes (how and when organizations should generate SBOMs)
- Automated: Software composition analysis (SCA)

https://www.ntia.doc.gov/report/2021/minimum-elements-software-bill-materials-sbom

SBOM Example


```
{
  "type": "library",
  "name": "electron",
  "version": "11.1.1",
  "bom-ref": "pkg:npm/electron@11.1.1",
  "author": "marshallofsound+electronhqnpm@electronjs.org, info@electronjs.org",
  "description": "Build cross platform desktop apps with JavaScript, HTML, and CSS",
  "licenses": [
    {
      "license": {
        "id": "MIT",
        "text": {
          "content": "Q29weXJpZ2h0IChjKSAyMDEzLTIwMjAgR2l0SHViIEluYy4KClBlcm1pc3Npb24gaXMgaGVyZWJ5IGdyYW50ZWQsIGZy
          "contentType": "text/plain",
          "encoding": "base64"
        }
      }
    }
  ],
  "copyright": "2013-2020 GitHub Inc",
  "purl": "pkg:npm/electron@11.1.1"
},
```

VEX

(Vulnerability Exploitability eXchange)

- Type of a security advisory
- Indicates whether a product or products are affected by known vulnerabilities.

Idea: companies can publish VEXs of their vulnerable software versions, other companies can check their SBOMs for these vulnerable versions and investigate impact (and then publish themselves if vulnerable and so on)

Recap: Objectives

- Outline common risks associated with code dependencies, including dependency confusion

Build Infrastructure

Trusting Trust

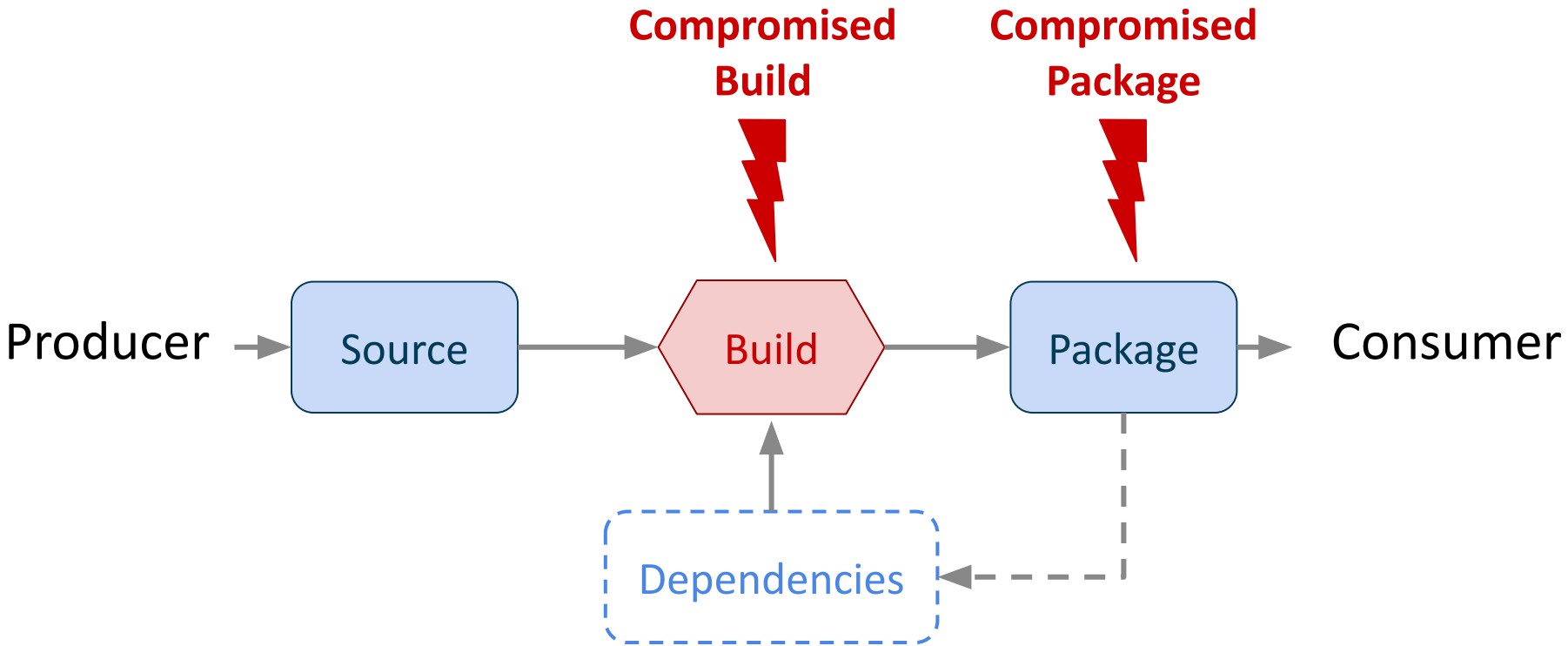
TURING AWARD LECTURE

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.

Ken Thompson

Build Attacks



Build Infrastructure

Build infrastructure can be as complex as the project code itself

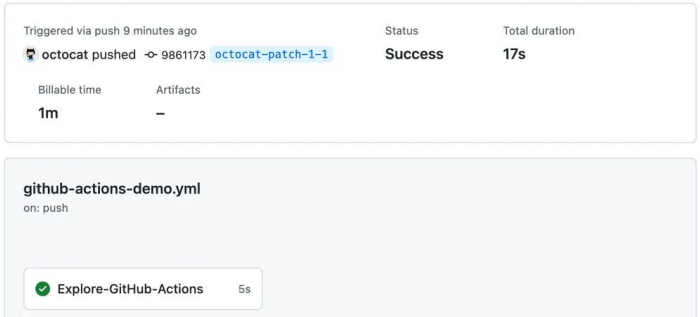
- Build tools, package managers, automation scripts
- Parallel ecosystem of code (e.g., GitHub Actions)
- Configuration with nested code and complex build, test, and deploy workflows

Codecov

DIVE BRIEF

Codecov hack — likened to SolarWinds — targets software supply chain

Published April 23, 2021 • Updated April 30, 2021



CI/CD Environments

Continuous Integration are dangerous Remote Code Execution engines

		TravisCI	CircleCI	Jenkins	Gitlab CI extrernal	Gitlab CI internal	Github CI
Admittance Control	(C1) Contributor can add workflow	●	●	●	●	●	●
	(C2) CI/CD run can add new workflow	○	○	●	○	○	●[w]
	(C3) Executes workflow from PR w/o merge	○	◑	●	○	◑	◑[w]
Execution Control	(C4) Contributors can modify the triggers	○	○	○	●	●	●
	(C5) CI/CD run can modify the triggers	○	○	●	○	○	●[w]
Code Control	(C6) CI/CD run can modify the code	○	○	●	○	○	●[w]
	(C7) CI/CD run can change behavior w/o modifying config	○	○	○	○	○	●[w]
Access to Secrets	(C8) Masked	●	●	◑	●	●	●
	(C9) Available to all steps	●	●	◑	○	○	●
	(C10) Available to pull requests	○	◑	◑	○	◑	◑[w]

GitHub Actions

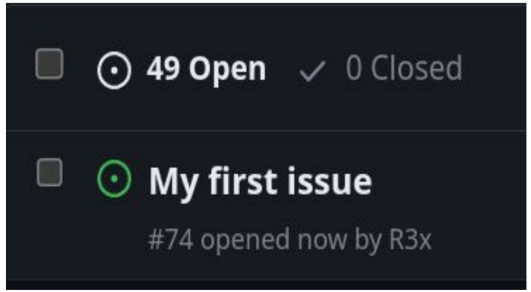
```yaml
name: Sample Workflow
on:
  issues:
    types: [opened]

jobs:
  notify:
    - name: Log title
      run: echo "${{ github.event.issue.title }}"
    ...
```

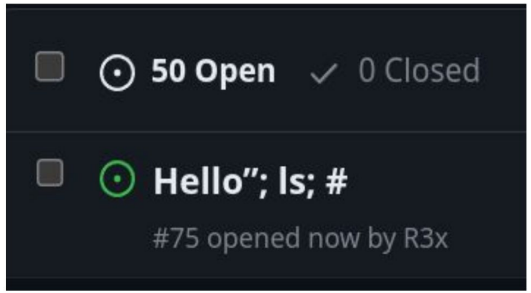
GitHub Actions

```yaml
name: Sample Workflow
on:
  issues:
    types: [opened]

jobs:
  notify:
    - name: Log title
      run: echo "${{ github.event.issue.title }}
    ...
```




```
echo “My first issue”
```



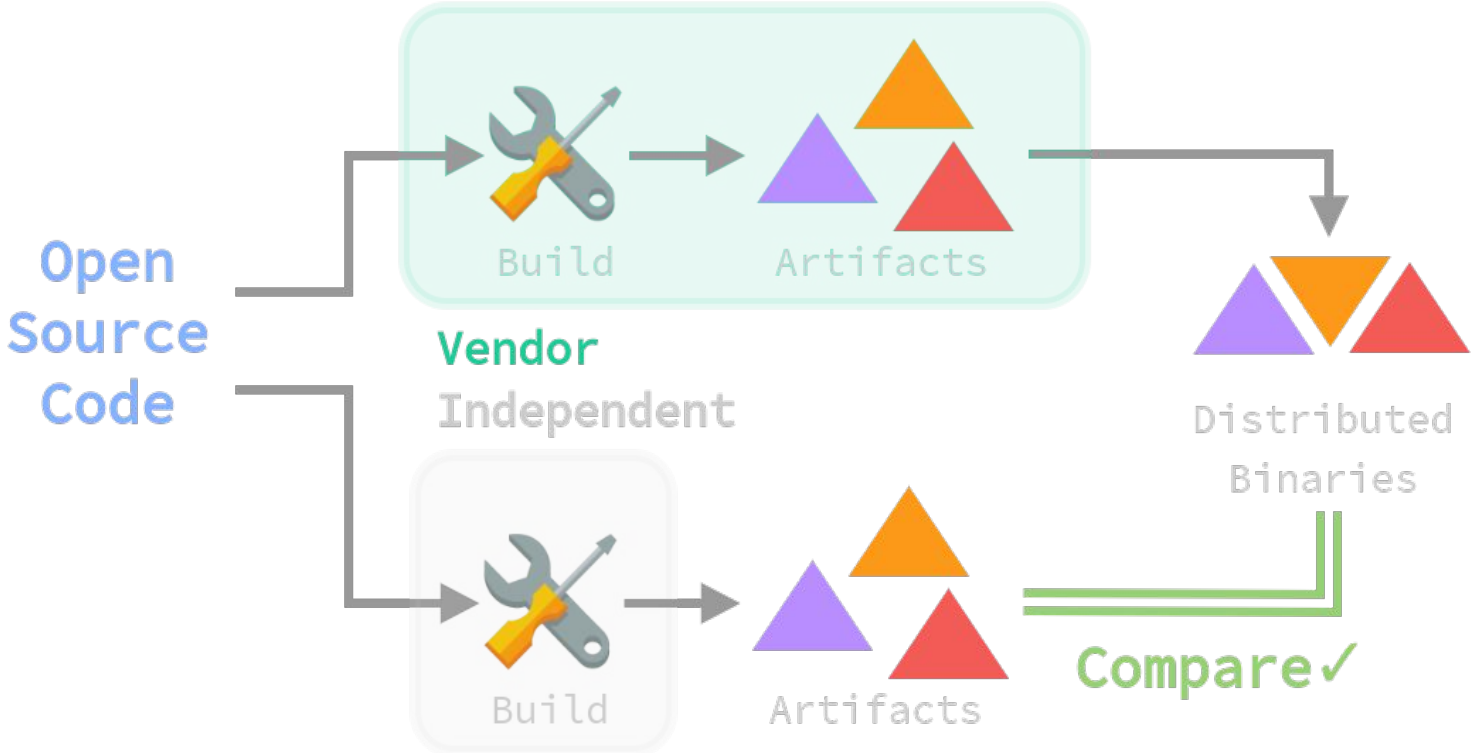

```
echo “Hello”; ls; #”
```

Attack: Compromise Build

- Run malicious build
- Tamper with build job
- Tamper with build system

How would you defend against these?

Defense: Reproducible Builds

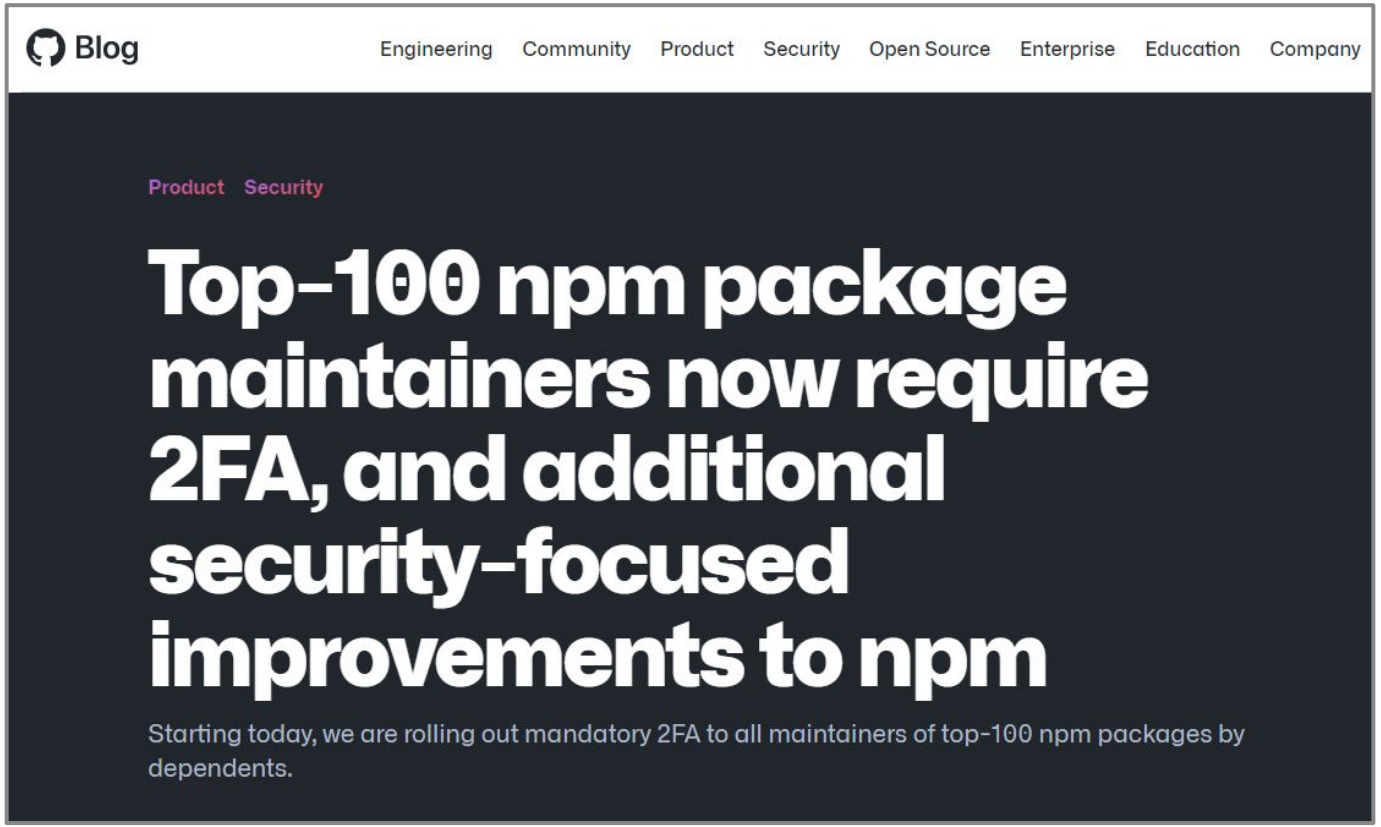


Like Flossing Your Teeth

Fourné et al. "It's like flossing your teeth: On the Importance and Challenges of Reproducible Builds for Software Supply Chain Security." IEEE S&P, 2023.

- Interviews with n=24 reproducible builds stakeholders from the mailing list
- Problems for RB: Build timestamps, build paths, filesystem ordering, archive metadata, randomness, uninitialized memory, …

Defense: Platform Hardening



Defense: Signing

Signing of commits, tags, build artifacts

- Supported by GitHub (and git)
- But might interfere with existing pipelines / workflows

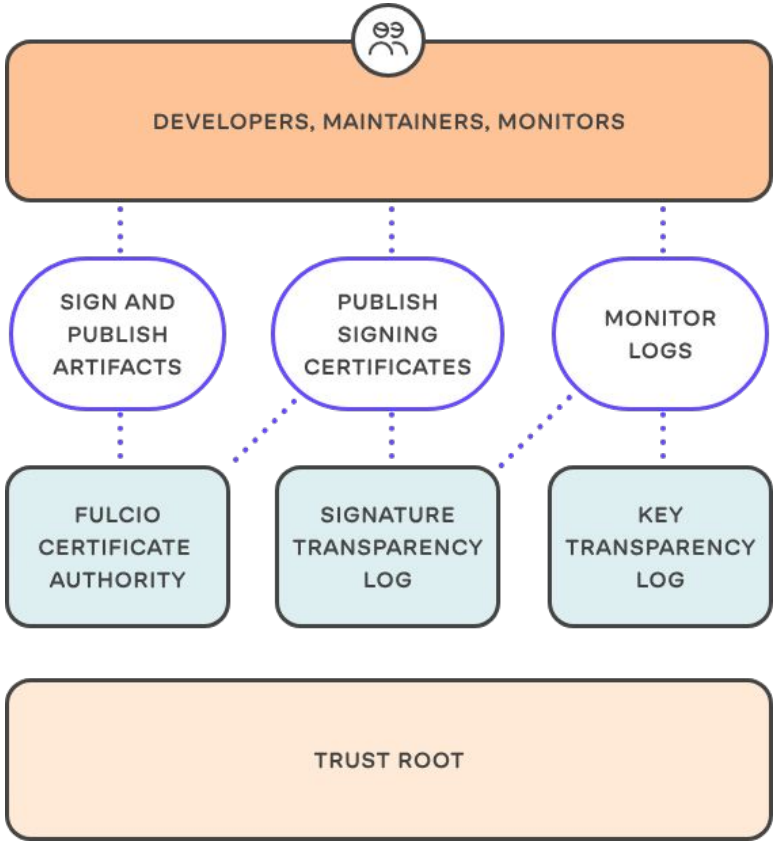

```
test-repo on  main ➜ git verify-commit HEAD
gpg: Signature made Wed 06 Mar 2024 10:55:58 PM EST
gpg:                using EDDSA key 7830813D924FF9A06FB0E6C
gpg:                issuer "git@dwermke.com"
gpg: Good signature from "Dominik Wermke <git@dwermke.com>"
```

Sigstore

https://www.sigstore.dev/

Idea:

- simplify (remove) key management
- integrate with package managers



SLSA

https://slsa.dev/

- Adoptable guidelines for supply chain security
- Multiple SLSA levels and (planned) tracks
- Provenance through SLSA attestations for artifacts
- Example: Build L3 requirement is a Hardened build platform

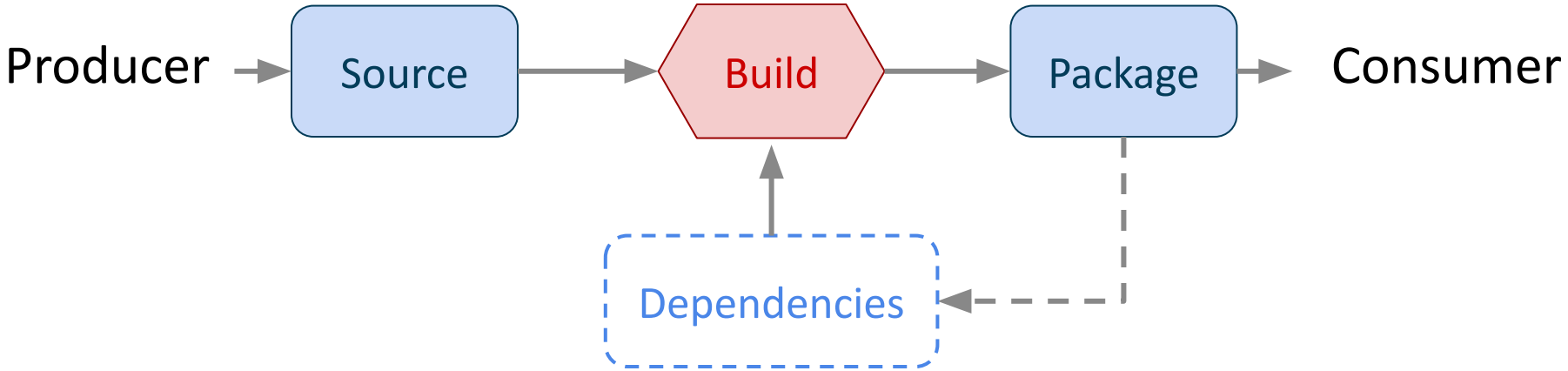
Recap: Objectives

- Outline good security practices for build infrastructure, including signing and reproducible builds.

The Human Factor

The Human Factor?

Where do you see the “Human Factor” in this SSC graph?



The Human Element

Humans as the weakest (?) link in the supply chain.

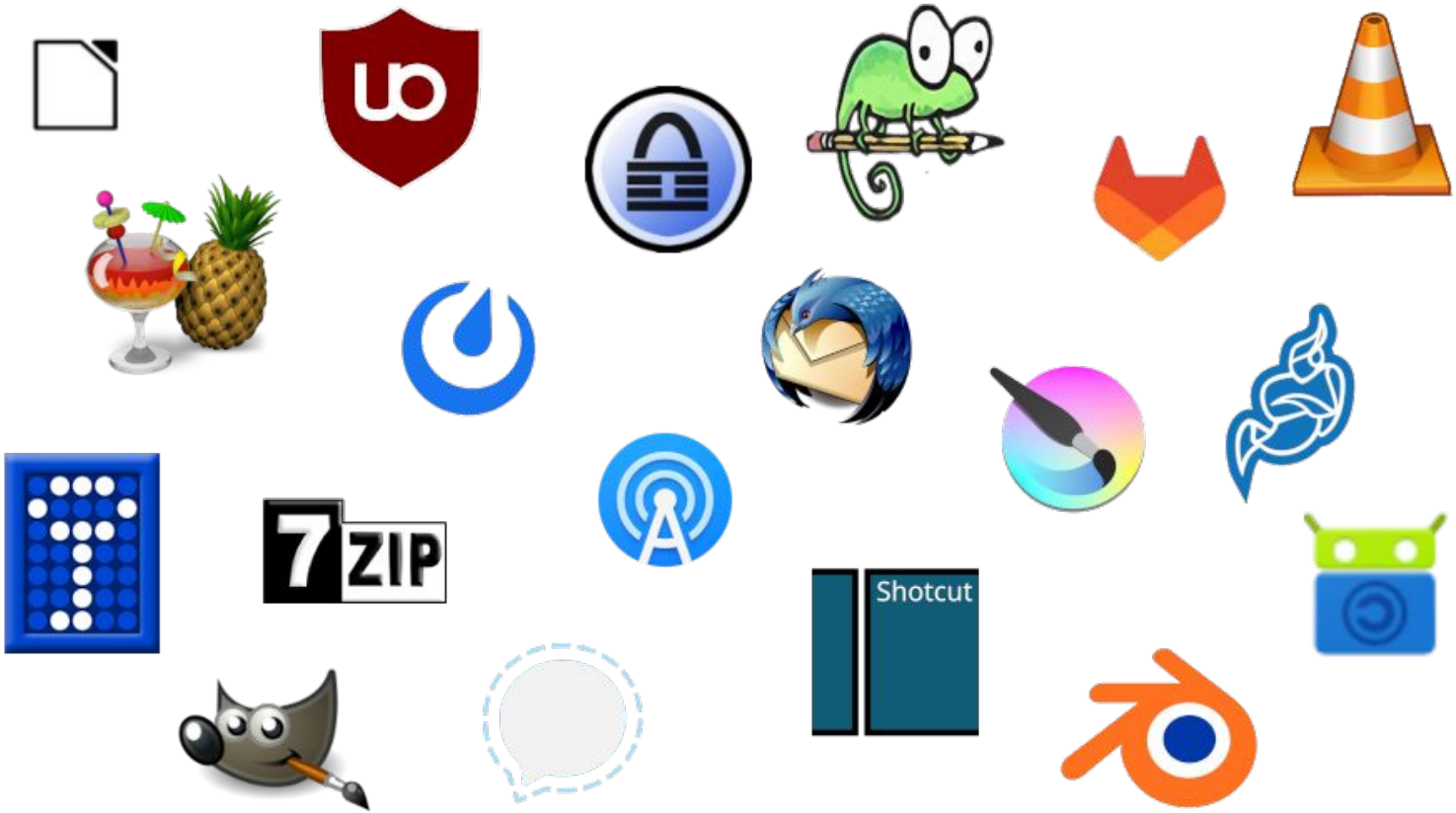
A Breach at LastPass Has Password Lessons for Us All

The hacking of the password manager should make us reassess whether to trust companies to store our sensitive data in the cloud.

New York Times

"The company said intruders had gained access to its cloud database and obtained a copy of the data vaults of tens of millions of customers by using credentials and keys stolen from a LastPass employee."

Open Source Software



Open Source Reuse

Open Source code appears as foundation, glue, or during the build process in many software projects.

xz-utils

XZ Utils

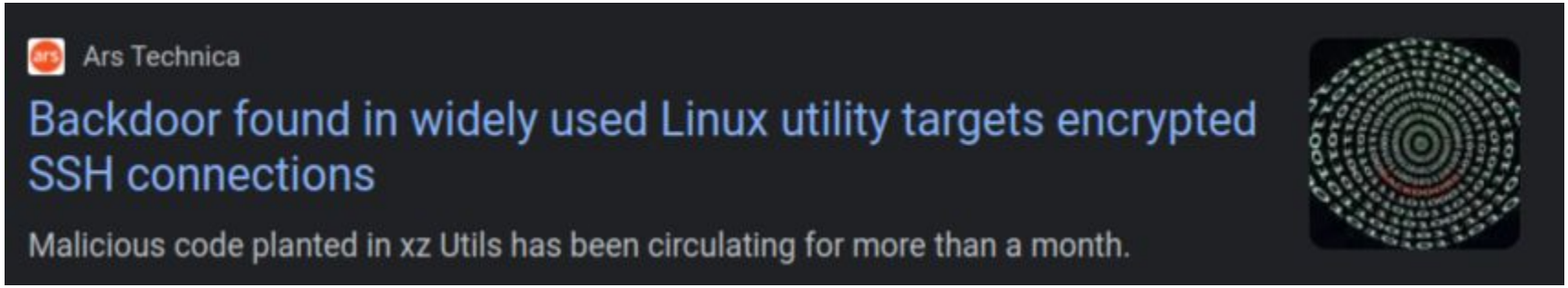
Documentation

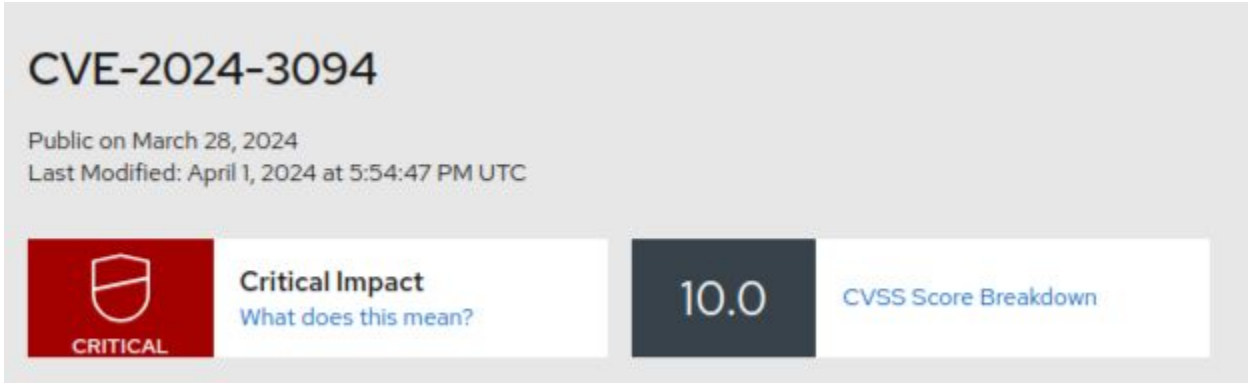
XZ Utils is a complete C99 implementation of the .xz file format. XZ Utils were originally written for POSIX systems, but has been ported to a few non-POSIX systems over the years.

XZ Utils consist of several components:

- **liblzma** is a compression library with an API similar to that of zlib.
- **xz** is a command line tool with syntax similar to that of gzip.
- **xzdec** is a decompression-only tool smaller than the full-featured xz tool.
- A set of shell scripts (**xzgrep**, **xzdiff**, etc.) have been adapted from gzip to ease viewing, grepping, and comparing compressed files.

xz Incident





https://research.swtch.com/xz-timeline

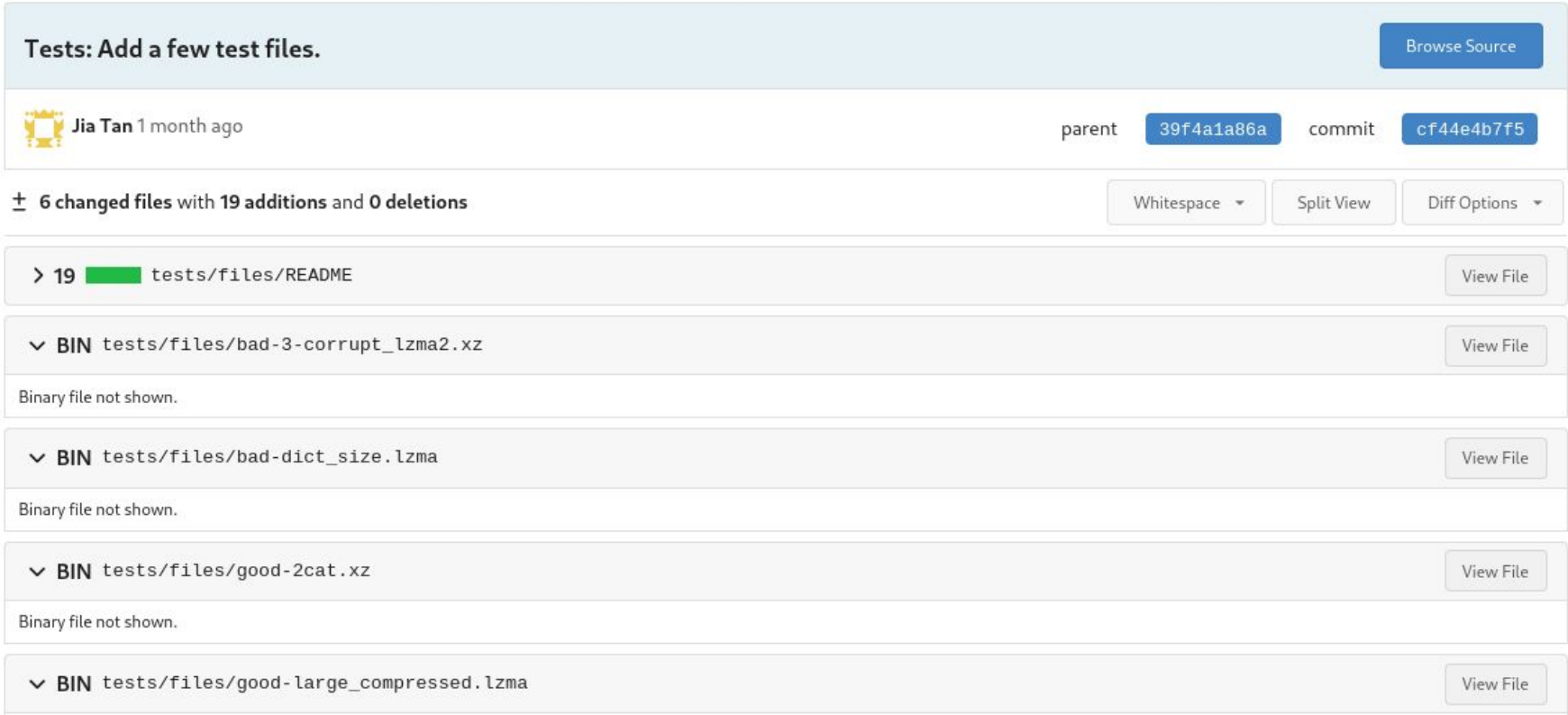
Discovery

I accidentally found a security issue while benchmarking postgres changes.

If you run debian testing, unstable or some other more "bleeding edge" distribution, I strongly recommend upgrading ASAP.

openwall.com/lists/oss-securit...

Exploit

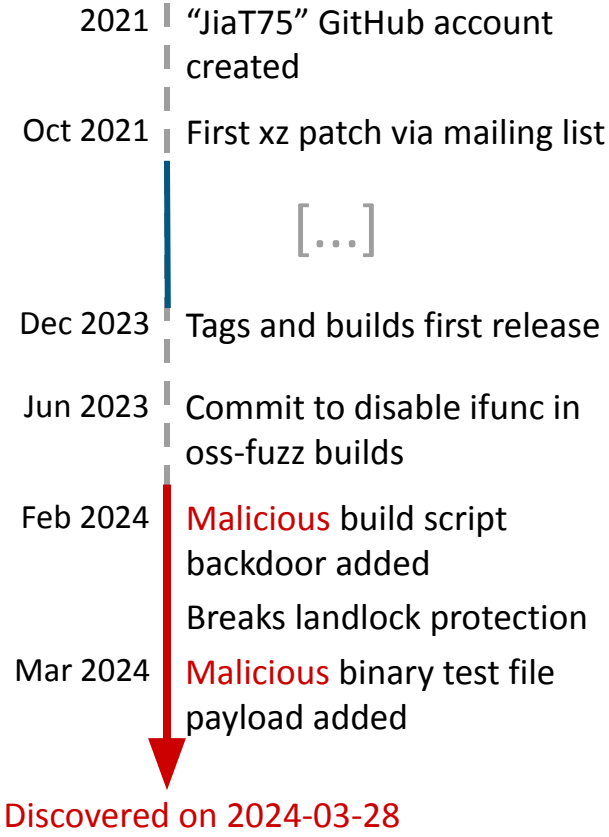

Tests: Add a few test files.
Browse Source
Jia Tan 1 month ago
parent 39f4a1a86a commit cf44e4b7f5
± 6 changed files with 19 additions and 0 deletions
Whitespace
Split View
Diff Options
19 tests/files/README
View File
BIN tests/files/bad-3-corrupt_lzma2.xz
View File
Binary file not shown.
BIN tests/files/bad-dict_size.lzma
View File
Binary file not shown.
BIN tests/files/good-2cat.xz
View File
Binary file not shown.
BIN tests/files/good-large_compressed.lzma
View File

Preparations

A Worst-Case Scenario

The xz-utils attack as (almost) worst case scenario:

- Attacker was (release) trusted developer on the project
- Multiple years of gaining trust and access and multiple months of setting up the backdoor
- Attacker manipulated external services (like oss-fuzz) to circumvent detection

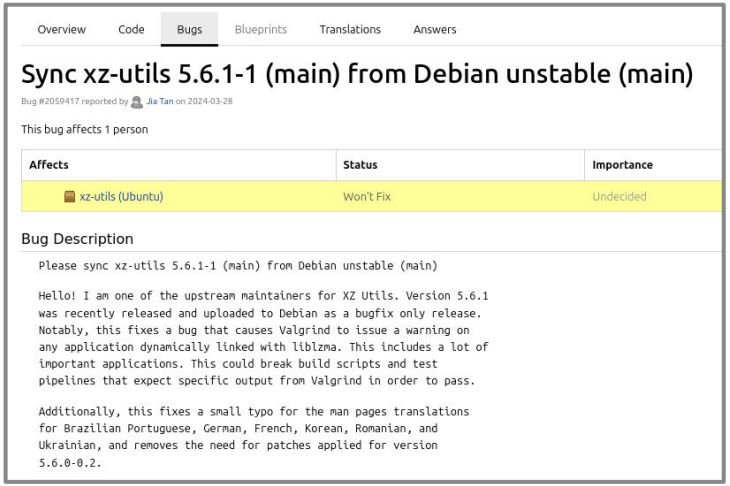


Humans as Attack Avenue

Re: [xz-devel] XZ for Java

mailto:jigarkumar17@protonmail.com?Subj

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.



Overview Code Bugs Blueprints Translations Answers

Sync xz-utils 5.6.1-1 (main) from Debian unstable (main)

Bug #2059417 reported by Jia Tan on 2024-03-28

This bug affects 1 person

Affects	Status	Importance
xz-utils (Ubuntu)	Won't Fix	Undecided

Bug Description

Please sync xz-utils 5.6.1-1 (main) from Debian unstable (main)

Hello! I am one of the upstream maintainers for XZ Utils. Version 5.6.1 was recently released and uploaded to Debian as a bugfix only release. Notably, this fixes a bug that causes Valgrind to issue a warning on any application dynamically linked with liblzma. This includes a lot of important applications. This could break build scripts and test pipelines that expect specific output from Valgrind in order to pass.

Additionally, this fixes a small typo for the man pages translations for Brazilian Portuguese, German, French, Korean, Romanian, and Ukrainian, and removes the need for patches applied for version 5.6.0-0.2.

Oct 2021	First xz patch via mailing list
Feb 2022	First merged commit
Apr 2022	**Pressure** on mailing list to merge patches
May 2022	“Is this maintained?” **pressure** on mailing list
Jun 2022	**Pressure** to add more maintainers on mailing list
Jun 2022	**Pressure** to add patch author (attacker) on mailing list
Dec 2023	Attacker tags and builds first release

Solitary Maintainer

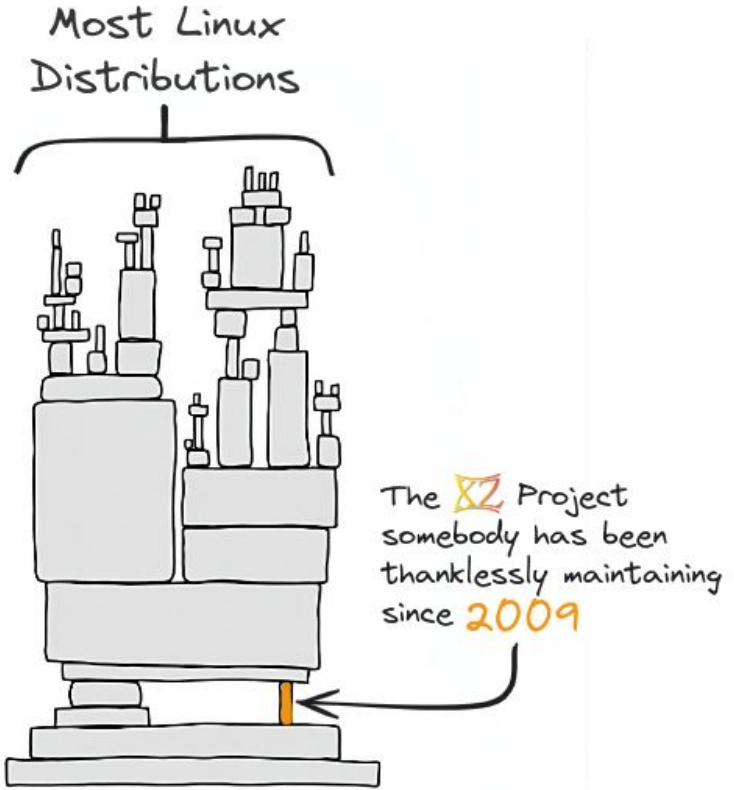
Challenge: Important open-source components maintained by solitary & unsupported hobby maintainers

https://foundation.rust-lang.org/grants/:

Hardship Grants

Awards ranging from $500 to $1,500 made to active maintainers of the Rust Project facing financial hardship.

>> Learn More



Adopted from xkcd 2347

SSC Frameworks

NIST Special Publication 800-218

Secure Software Development Framework (SSDF) Version 1.1:

Recommendations for Mitigating the Risk of Software Vulnerabilities

NIST Special Publication
NIST SP 800-161r1

Cybersecurity Supply Chain Risk Management Practices for Systems and Organizations

OpenSSF
Secure Supply Chain Consumption Framework

SLSA

OWASP SCVS
Software Component Verification Standard
Measure and Improve Software Supply Chain Assurance

Software Supply Chain Best Practices

CLOUD NATIVE COMPUTING FOUNDATION

BSIMM 13

P-SSCRM

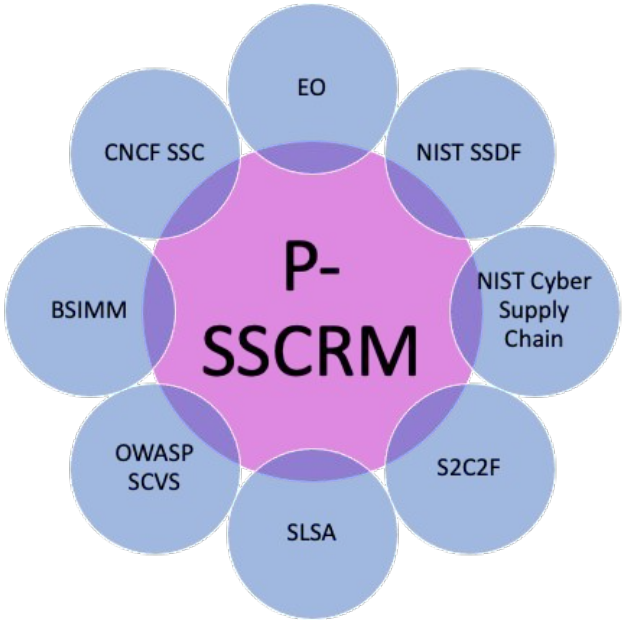
Idea: Union of SSC frameworks

Goal: Guiding companies towards better supply chain security

Implementation: Proactive Software Supply Chain Risk Management (P-SSCRM) framework

Model: 73 risk management tasks

Proactive Software Supply Chain Risk Management Framework (P-SSCRM) Version 1.0



Commercial Platforms

There are many (commercial) platforms out there for software supply chain security.

How to Support Open Source?

Frameworks are more targeted towards industry, commercial platforms cost money.

How to support software supply chain security in open source projects?

Open Source Security Foundation

https://openssf.org/

- Community of developers (some from big companies)
- Goal: secure open source software for the greater public good

Open Problems

1. How to meaningfully engage with supply chain stakeholders?
2. How to improve adoption of effective technical solutions?
3. How to measure improvements?

Learning Objectives

- Describe key challenges for securing the software supply chain in the context of humans.

Recap of Today

1. Intro: (Software) Supply Chain
2. SSC Areas:
    a. Code Dependencies
    b. Build Infrastructure
    c. The Human Factor

Discussion Paper

[DISC] Cox, Fifty Years of Open Source Software Supply Chain Security: For decades, software reuse was only a lofty goal. Now it's very real., ACM Queue, 2025.

Do you think

1. Open source reuse outpaced security practices?
2. Governance is required beyond tooling to fix OSS supply-chain risk?
3. Reproducible builds are realistic at scale?