Bytes, Pipes, and People

Seth Michael Larson

https://sethmlarson.dev
@sethmlarson

Blogging about Python,
the internet, and security

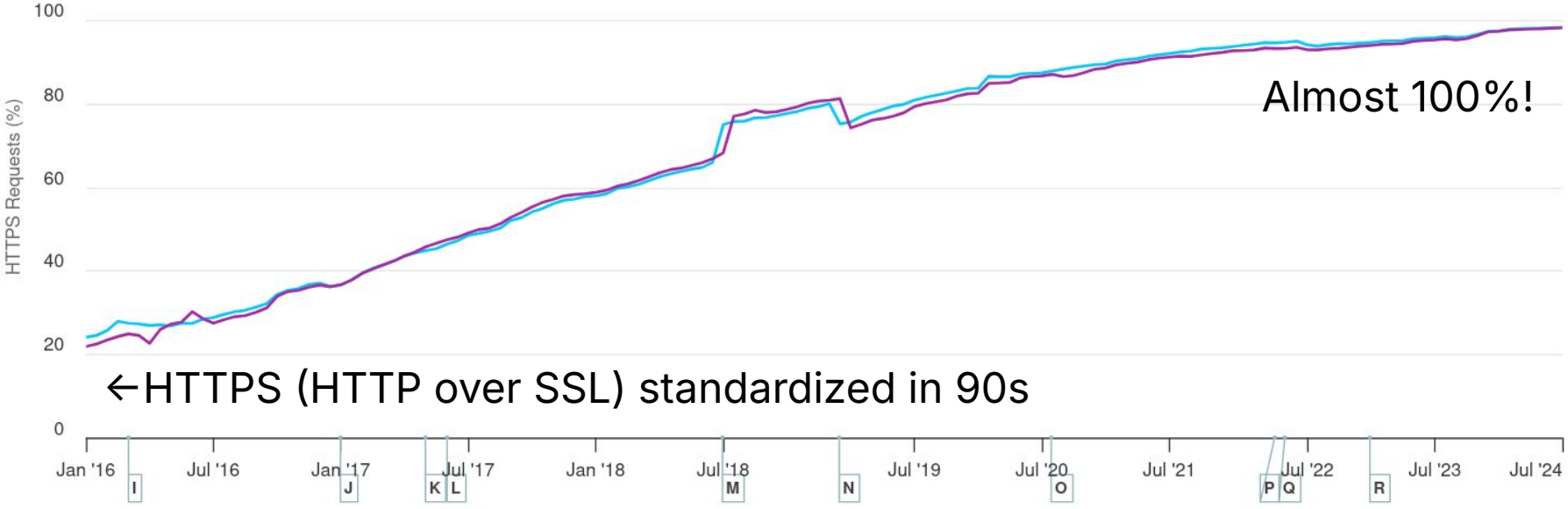
Slides and links! →

Securing the Web

HTTPS: Massive improvement to digital security!

Securing the Web

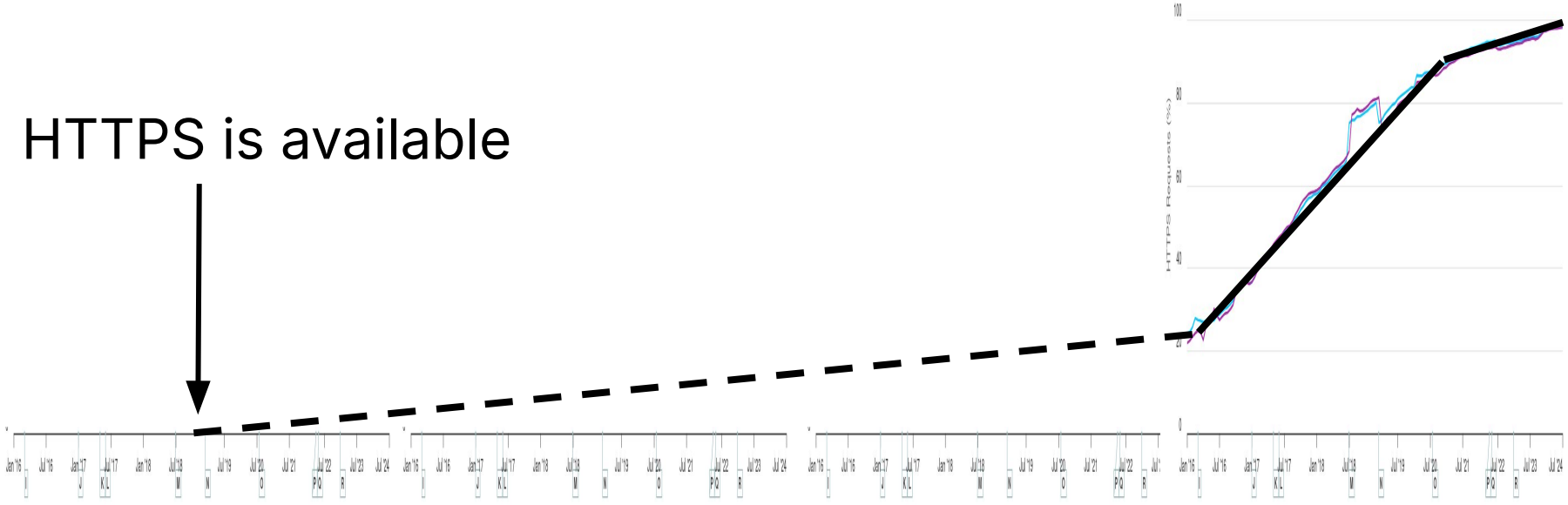
HTTPS: Massive improvement to digital security!



Securing the Web

HTTPS: Massive improvement to digital security!

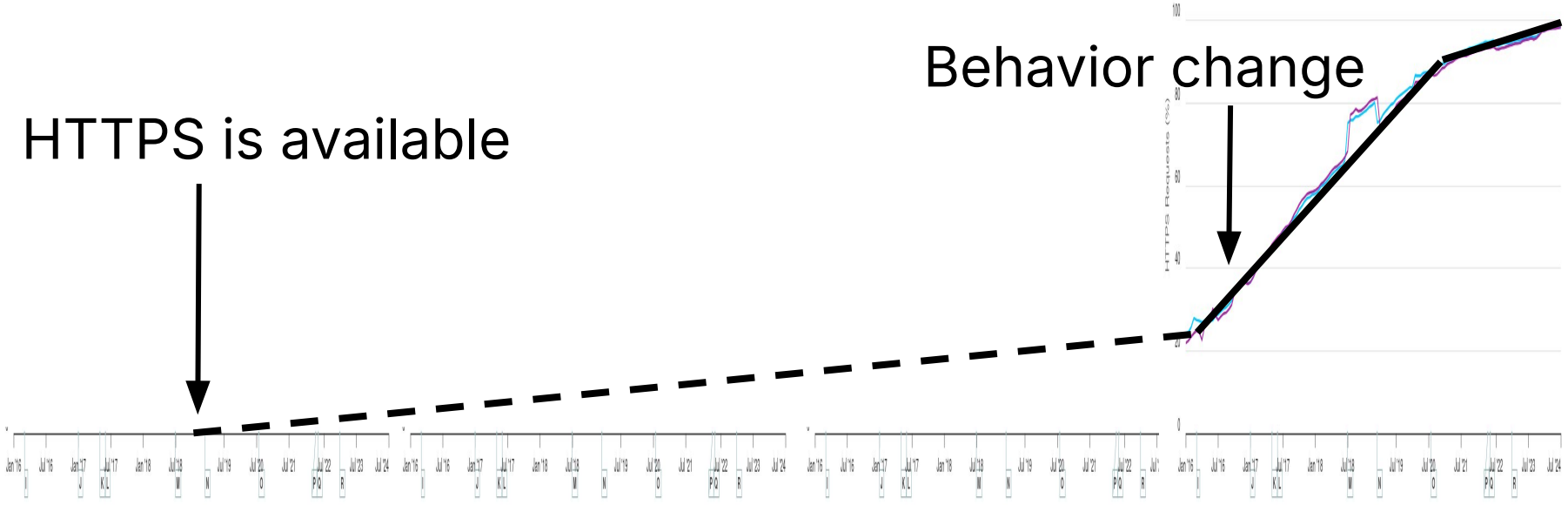
Needed **something else** to see **widespread adoption.**



Securing the Web

HTTPS: Massive improvement to digital security!

Needed **something else** to see **widespread adoption.**



Security Developer-in-Residence

Mission: Improve the security posture of Python, pip, and the **entire Python ecosystem.**

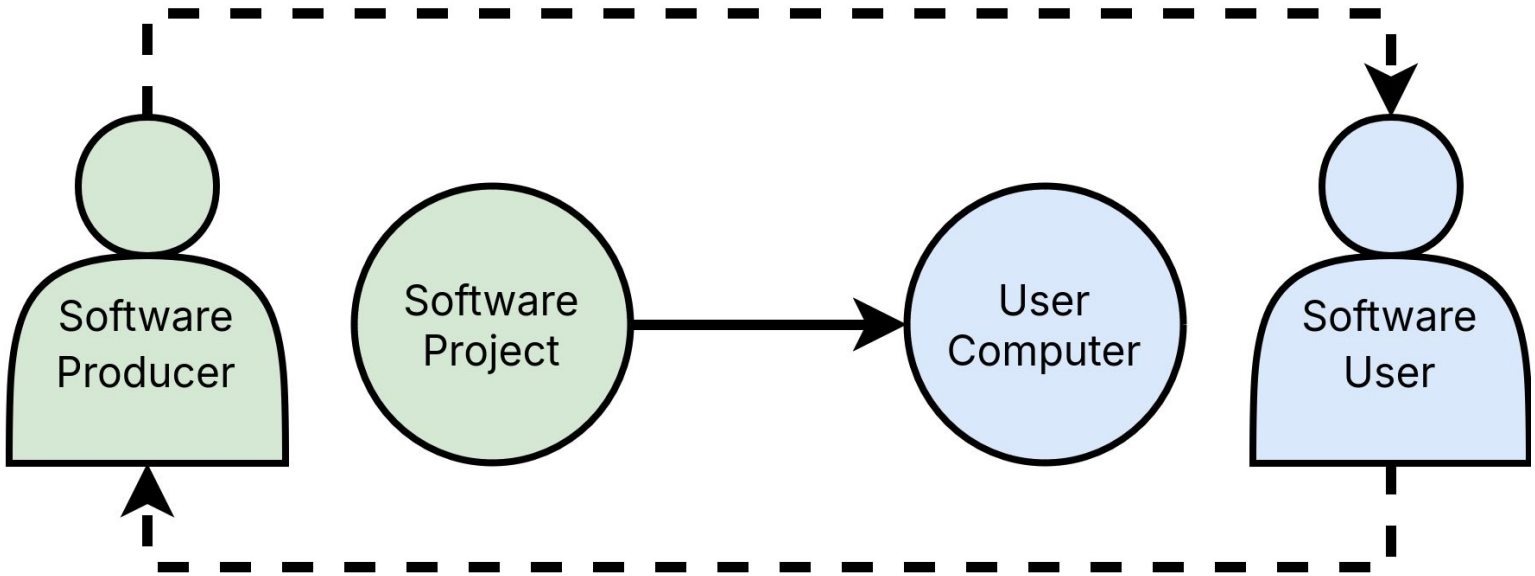
>600,000 packages on Python Package Index
Millions of Python programmers worldwide

Open Source Security Tabletop
SOSS Community Day 2024

Stories change Culture:

Behavior, institutions, and norms of groups

Positive Feedback Loop 📈



What to expect:

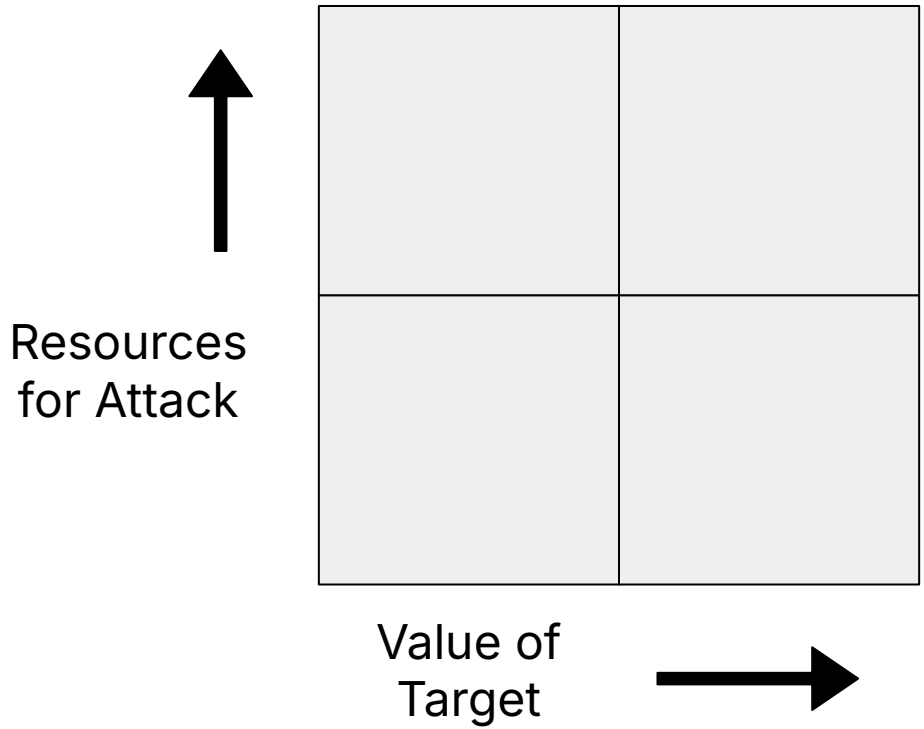
- Why is **open source security** important?
- Supporting a **culture of security** in Python with todays tools, data, and infrastructure
- How **everyone** can take action today to do their part for Python security

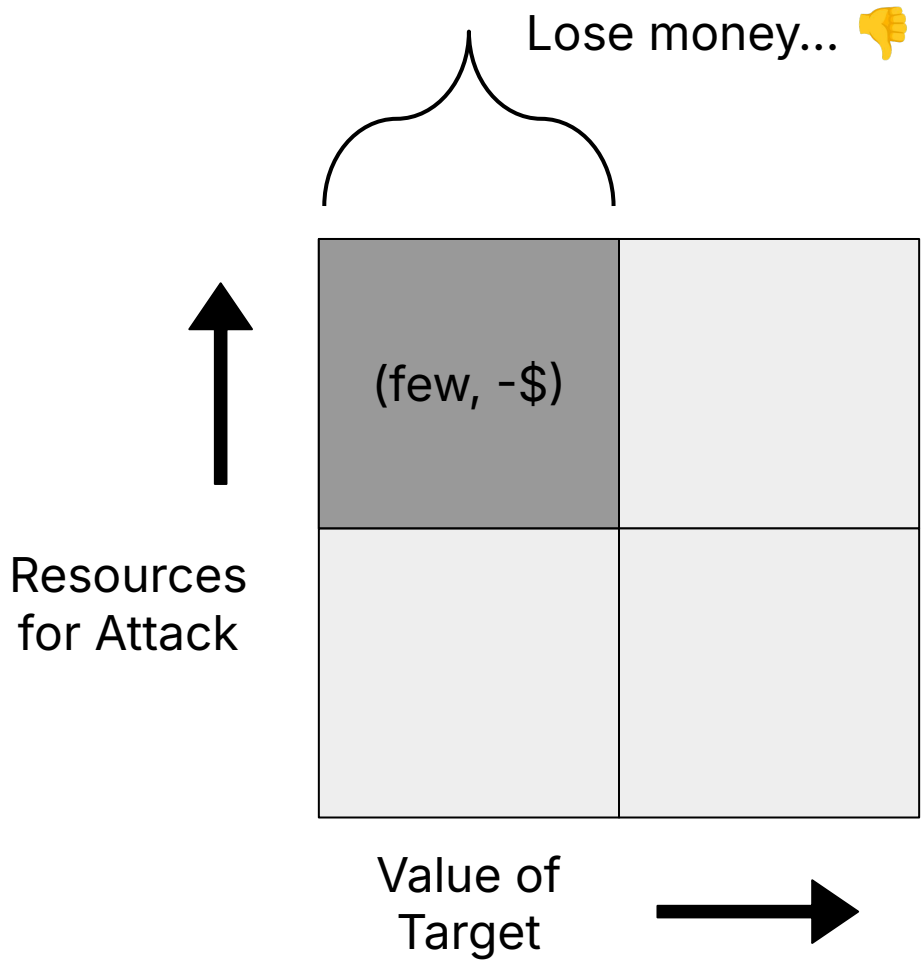
Let’s talk security (without the fear)

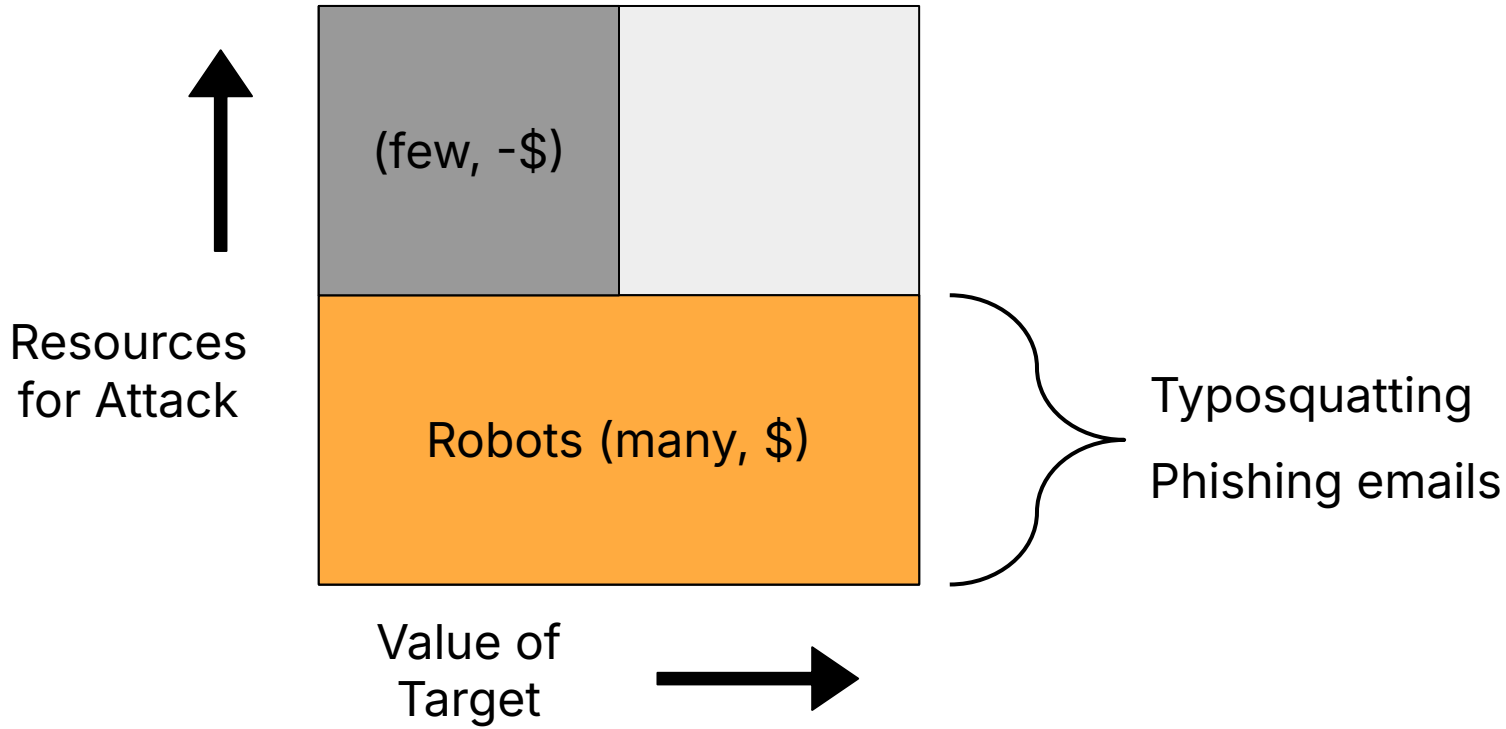
“Vuln Together” open space
PyCon US 2024

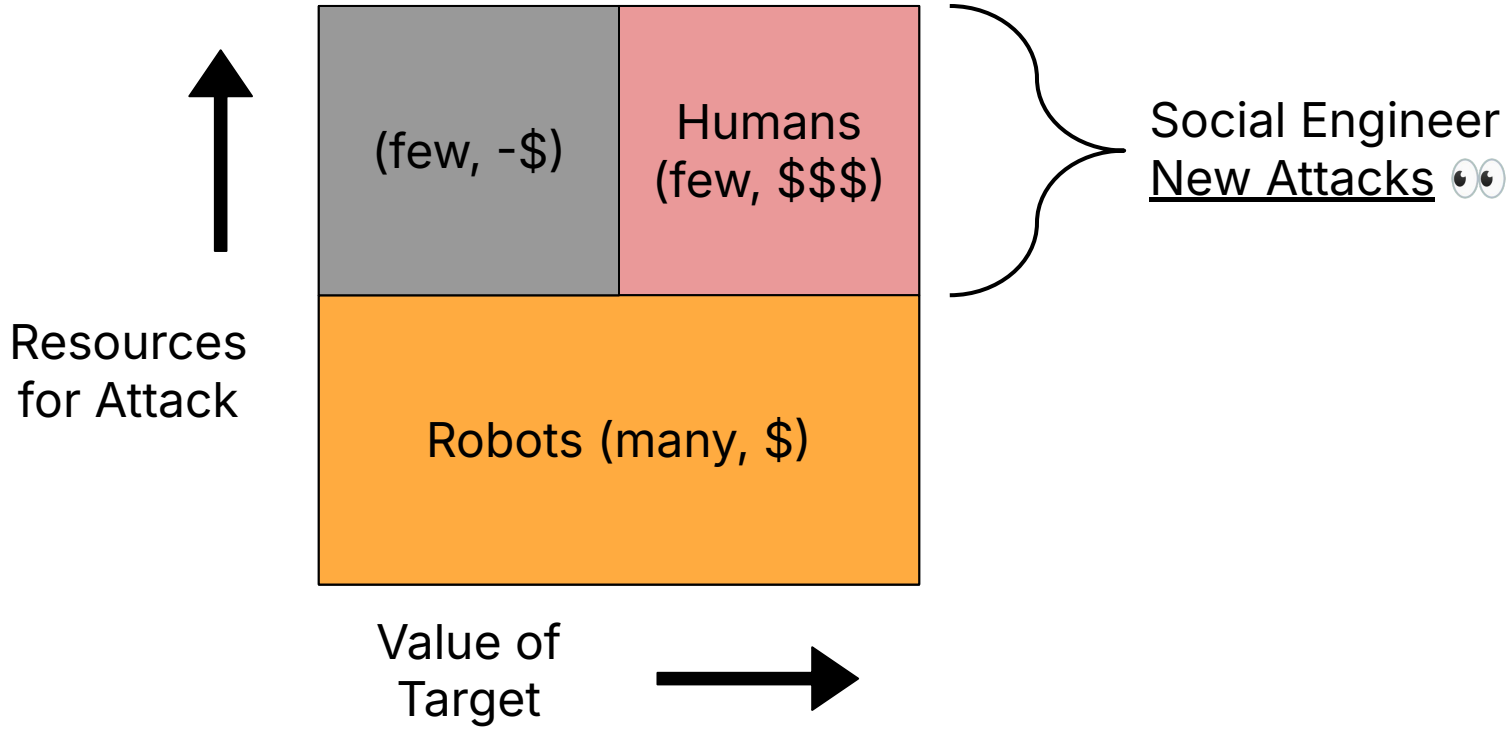
Chapter 0: Prologue

Open source security, briefly


Resources
for Attack
Value of
Target

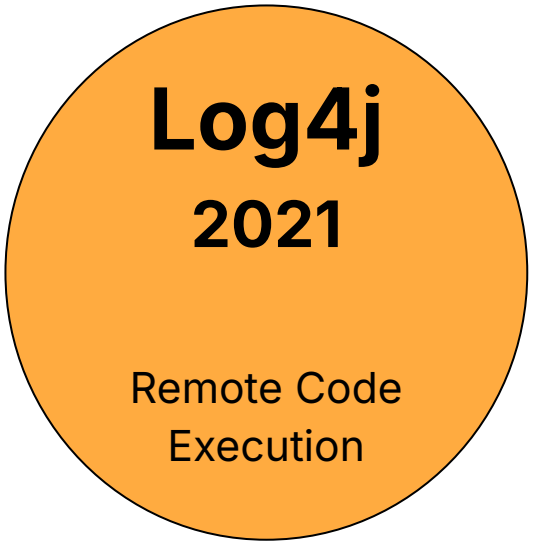

Lose money... 👎
(few, -$)
Resources for Attack
Value of Target


(few, -$)
Resources for Attack
Robots (many, $)
Typosquatting
Phishing emails
Value of Target

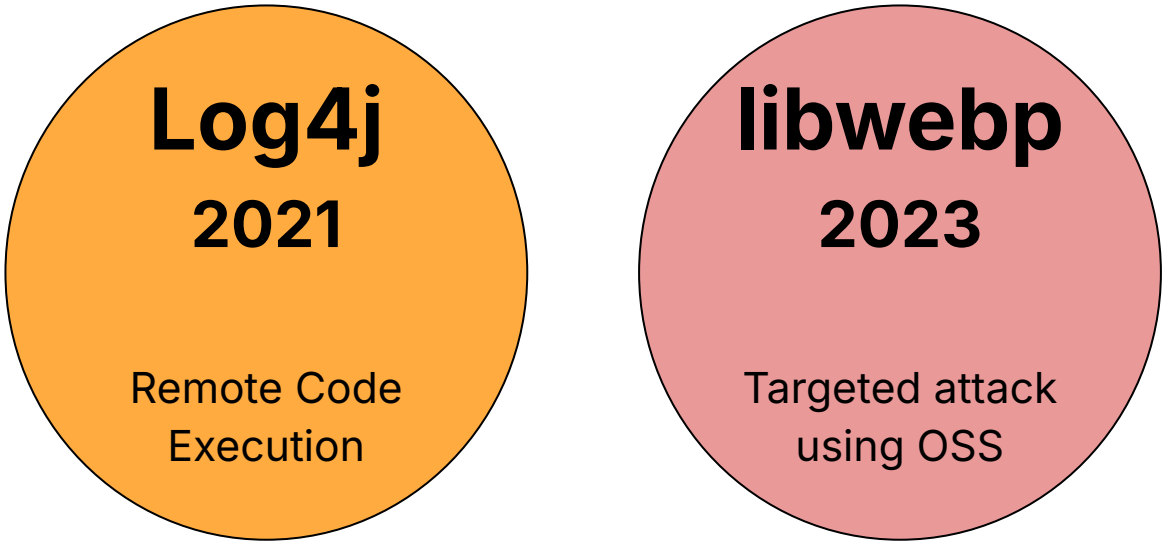

(few, -$)
Humans
(few, $$$)
Robots (many, $)
Social Engineer
New Attacks 👀
Resources
for Attack
Value of
Target

Why is software security important for open source?

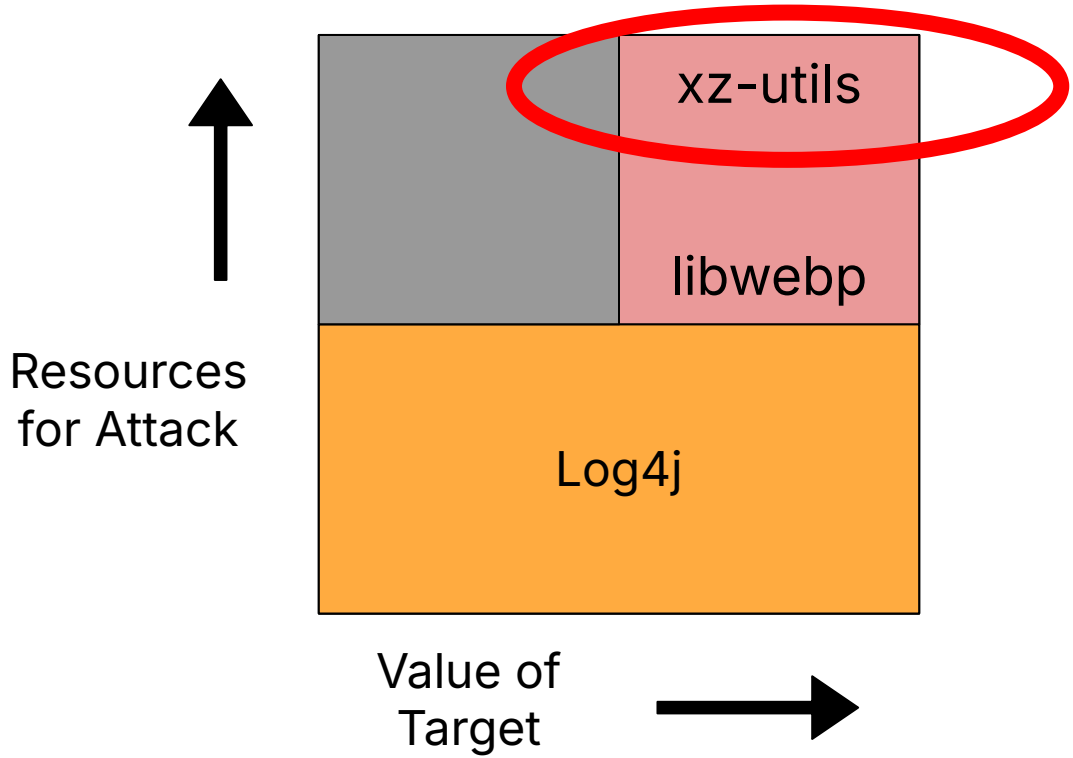
Why is software security important for open source?



Why is software security important for open source?



Why is software security important for open source?


xz-utils
libwebp
Log4j
Resources
for Attack
Value of
Target

Why is software security important for open source?

Why is software security important for open source?

>90%

Commercial software projects include open source components

Why is software security important for open source?

>90%

Commercial software projects include open source components

>70%

Lines of code in projects are open source components

Why is ~~software~~ security important for ~~open source~~?

>90%

Commercial software projects include open source components

>70%

Lines of code in projects are open source components

Why is open source security important for all software?

>90%

Commercial software projects include open source components

>70%

Lines of code in projects are open source components

Why is open source security important for all software?

MAY 12, 2021

Executive Order on Improving the Nation's Cybersecurity

BRIEFING ROOM › PRESIDENTIAL ACTIONS

By the authority vested in me as President by the Constitution and the laws of the United States of America, it is hereby ordered as follows:

Section 1. Policy. The United States faces persistent and increasingly sophisticated malicious cyber campaigns that threaten the public sector, the private sector, and ultimately the American people's security and privacy. The

Brussels, 15.9.2022
COM(2022) 454 final

2022/0272 (COD)

Proposal for a

REGULATION OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL

on horizontal cybersecurity requirements for products with digital elements and amending Regulation (EU) 2019/1020

(Text with EEA relevance)

Why is open source security important for all software?

White House
Executive Order 14028

Executive Order on Improving the Nation's Cybersecurity

BRIEFING ROOM › PRESIDENTIAL ACTIONS

By the authority vested in me as President by the Constitution and the laws of the United States of America, it is hereby ordered as follows:

Section 1. Policy. The United States faces persistent and increasingly sophisticated malicious cyber campaigns that threaten the public sector, the private sector, and ultimately the American people's security and privacy. The

EU Cyber
Resilience Act (CRA)

Brussels, 15.9.2022
COM(2022) 454 final

2022/0272 (COD)

Proposal for a

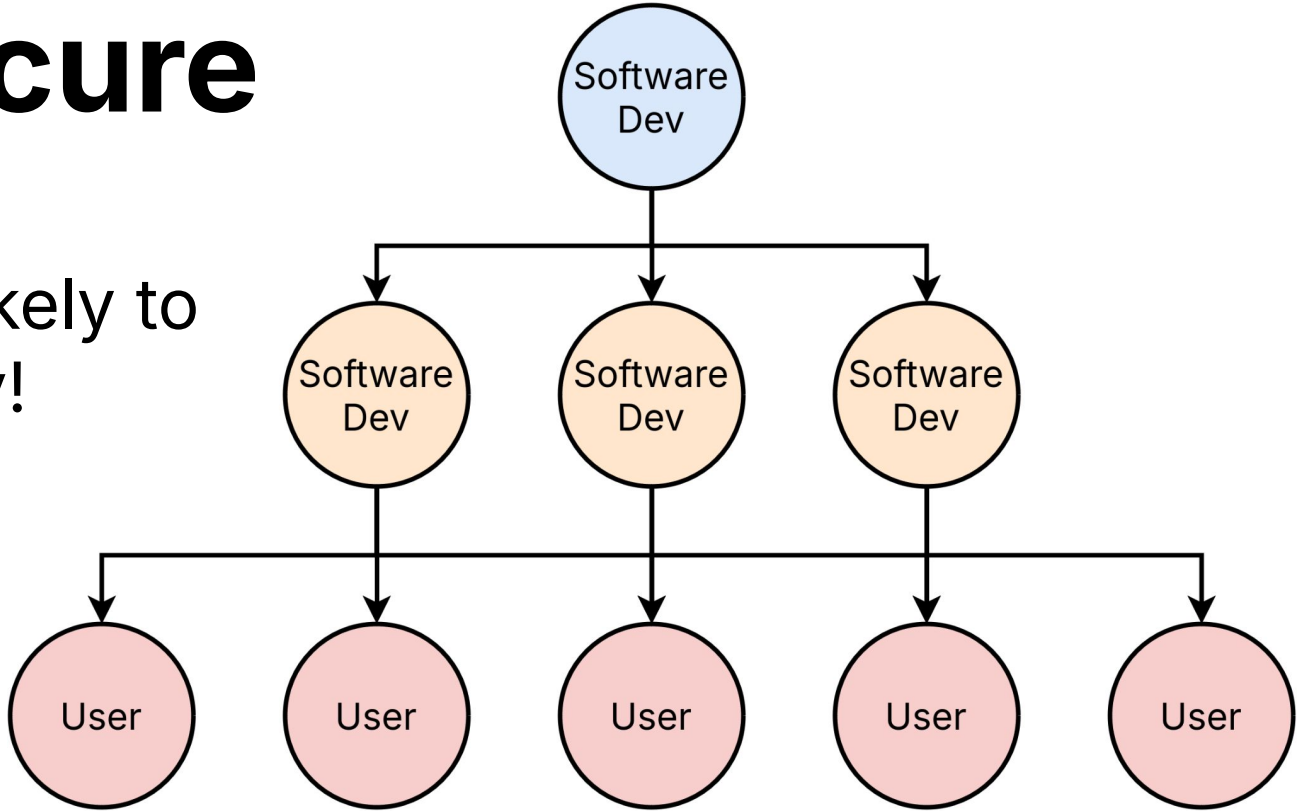
REGULATION OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL

on horizontal cybersecurity requirements for products with digital elements and amending Regulation (EU) 2019/1020

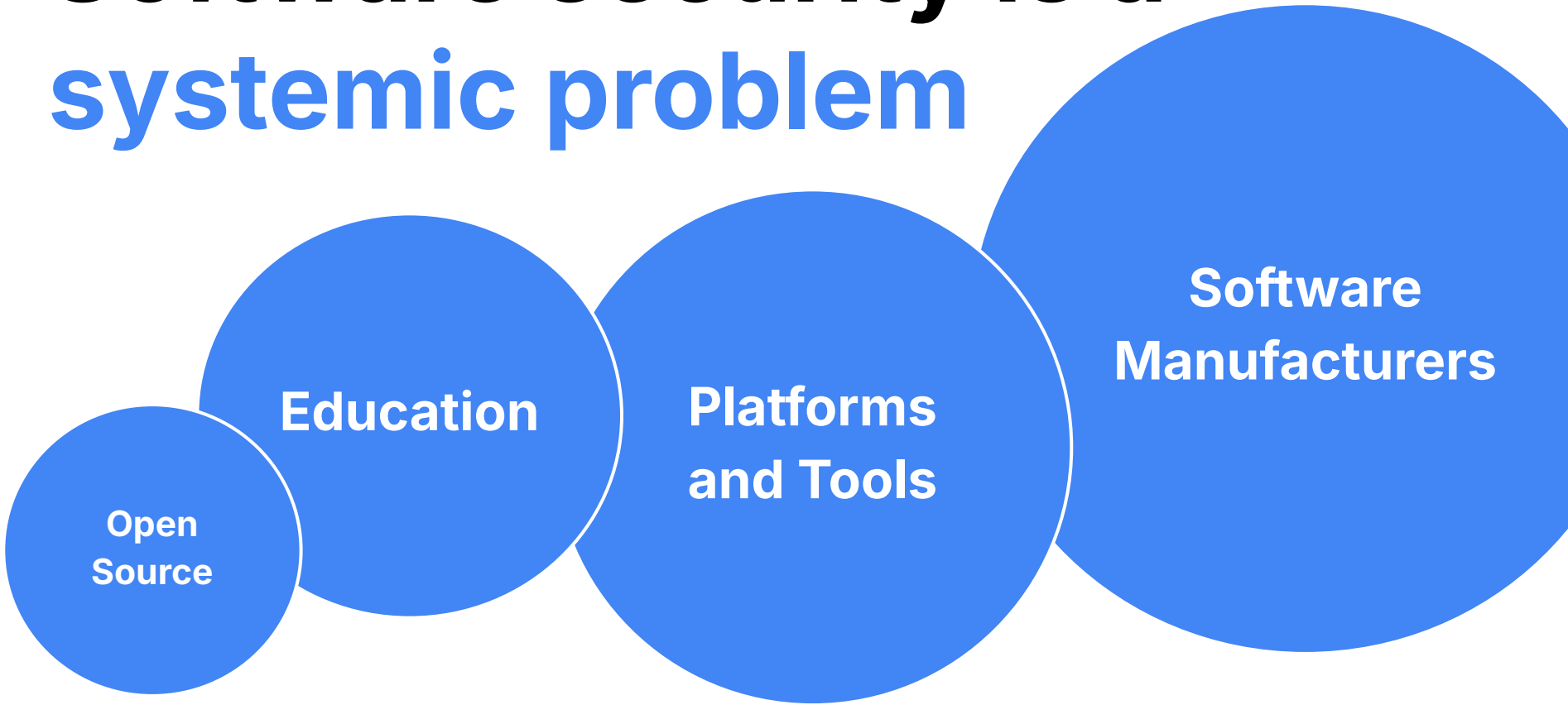
(Text with EEA relevance)

Users expect software to be secure

Users are not likely to ***ask*** for security!



Software security is a systemic problem



Chapter 1: Bytes

What is software made of?

Measuring software is hard

>> What software is running on this system?

- Applications
- Dependencies
- Versions
- Package Managers
- Copy-Paste, Patches
- Compiled Binaries
- Kernel, Firmware

¯_(ツ)_/¯

Software Bill-of-Materials

- Abbreviated: **SBOM** (pronounced: “S-bom”)

Software Bill-of-Materials

- Abbreviated: **SBOM** (pronounced: “S-bom”)
- “List of ingredients” for software 🎂
- **Directly referenced by White House and EU**

Software Bill-of-Materials

Situation: There are at least 2 competing standards

Projects creating SBOMs:

Choose one standard. Use JSON, not XML or tag:value

Most Python packages won't have to do this!

What is a vulnerability?

Confidentiality: Make secret data public

Integrity: Modify or delete data

Availability: Service not available to users

Tracked using many systems, CVE is most common!

Vulnerability scanners: SBOM → List of CVEs

What is a CVE?

“CVE-{YEAR}-{NUMBER}” → “CVE-2024-12345”

Free and shareable ID tracking a vulnerability

Description, affected project and versions, severity.

CVE is a pretty good system

Established infrastructure (since 1999)

Works great for announcing security fixes!

Friendly faces for open source projects:

Python Software Foundation, Red Hat, GitHub

Python provide SBOMs!

python SOFTWARE FOUNDATION

News from the Python Software Foundation

Thursday, February 08, 2024

Software Bill-of-Materials documents are now available for CPython

Our Security Developer-in-Residence, Seth Larson, has been working to improve the management of vulnerabilities for Python users. Seth has championed progress on this goal in a variety of areas:

- Authorizing the Python Software Foundation as a CVE Numbering Authority (CNA) to publish CVE IDs and records
- Revitalizing the security advisory mailing list (security-announce@python.org)
- Migrating all historical vulnerabilities to the Open Source Vulnerability

SPDX®

Available CPython 3.12.2+

Software Bill-of-Materials and CVE in action!

Files

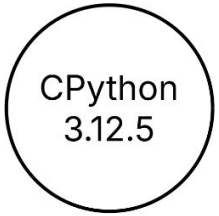
Version	Operating System	Description	MD5 Sum	File Size	GPG	Sigstore	SBOM
Gzipped source tarball	Source release		d23d56b51d36a9d51b2b13d30c849d00	25.7 MB	SIG	.sigstore	SPDX
XZ compressed source tarball	Source release		02c7d269e077f4034963bba6befdc715	19.5 MB	SIG	.sigstore	SPDX
macOS 64-bit universal2 installer	macOS	for macOS 10.9 and later	78bd8d0795062b1df63e2b8d8386a5fa	43.5 MB	SIG	.sigstore	
Windows installer (64-bit)	Windows	Recommended	bbcb2fcf9d739f776fb6414afc12c80d	25.3 MB	SIG	.sigstore	SPDX
Windows installer (32-bit)	Windows		d151f5f116e11c4d40021527f51ddf67	24.0 MB	SIG	.sigstore	SPDX
Windows installer (ARM64)	Windows	Experimental	365d59eff83dfea9af528df4ebd060cb	24.5 MB	SIG	.sigstore	SPDX
Windows embeddable package (64-bit)	Windows		0f53697bdcecfb97b99ac8aa9d9a9e13	10.5 MB	SIG	.sigstore	SPDX
Windows embeddable package (32-bit)	Windows		82dd15b14c307f5fcef80ccb45d6b404	9.4 MB	SIG	.sigstore	SPDX
Windows embeddable package (ARM64)	Windows		62c81364c232644f280b06ef5f33a029	9.8 MB	SIG	.sigstore	SPDX

https://python.org/downloads

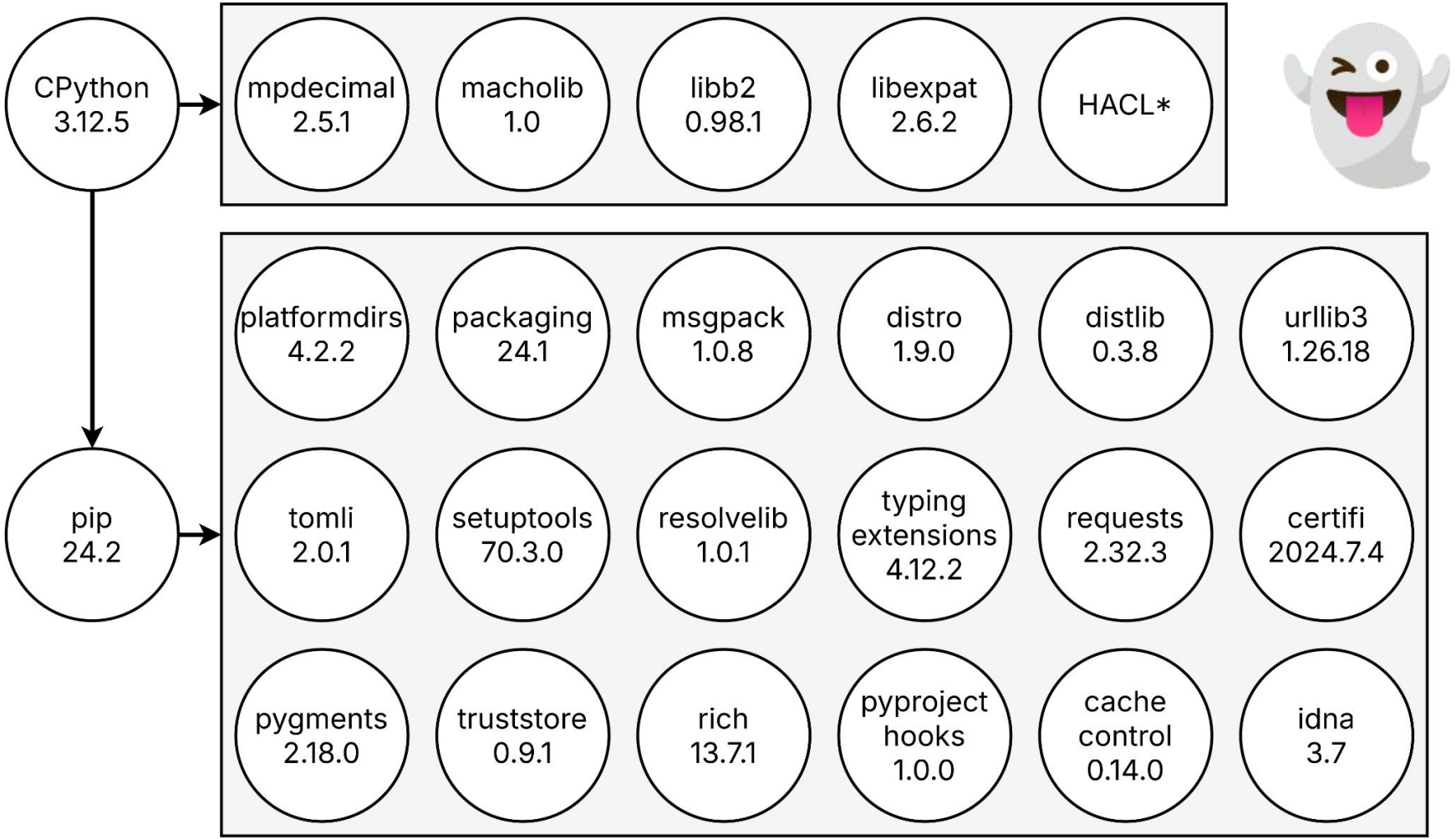
Files

Version	Operating System	Description	MD5 Sum	File Size	GPG	Sigstore	SBOM
Gzipped source tarball	Source release		d23d56b51d36a9d51b2b13d30c849d00	25.7 MB	SIG	.sigstore	SPDX
XZ compressed source tarball	Source release		02c7d269e077f4034963bba6befdc715	19.5 MB	SIG	.sigstore	
macOS 64-bit universal2 installer	macOS	for macOS 10.9 and later	78bd8d0795062b1df63e2b8d8386a5fa	43.5 MB	SIG	.sigstore	
Windows installer (64-bit)	Windows	Recommended	bbcb2fcf9d739f776fb6414afc12c80d	25.3 MB	SIG	.sigstore	SPDX
Windows installer (32-bit)	Windows		d151f5f116e11c4d40021527f51ddf67	24.0 MB	SIG	.sigstore	SPDX
Windows installer (ARM64)	Windows	Experimental	365d59eff83dfea9af528df4ebd060cb	24.5 MB	SIG	.sigstore	SPDX
Windows embeddable package (64-bit)	Windows		0f53697bdcecfb97b99ac8aa9d9a9e13	10.5 MB	SIG	.sigstore	SPDX
Windows embeddable package (32-bit)	Windows		82dd15b14c307f5fcef80ccb45d6b404	9.4 MB	SIG	.sigstore	SPDX
Windows embeddable package (ARM64)	Windows		62c81364c232644f280b06ef5f33a029	9.8 MB	SIG	.sigstore	SPDX

https://python.org/downloads


CPython
3.12.5


CPython
3.12.5
mpdecimal
2.5.1
macholib
1.0
libb2
0.98.1
libexpat
2.6.2
HACL*


CPython
3.12.5
mpdecimal
2.5.1
macholib
1.0
libb2
0.98.1
libexpat
2.6.2
HACL*
pip
24.2
platformdirs
4.2.2
packaging
24.1
msgpack
1.0.8
distro
1.9.0
distlib
0.3.8
urllib3
1.26.18
tomli
2.0.1
setuptools
70.3.0
resolvelib
1.0.1
typing
extensions
4.12.2
requests
2.32.3
certifi
2024.7.4
pygments
2.18.0
truststore
0.9.1
rich
13.7.1
pyproject
hooks
1.0.0
cache
control
0.14.0
idna
3.7

Vulnerability scanners

pip-audit: Pure Python scanning!

grype

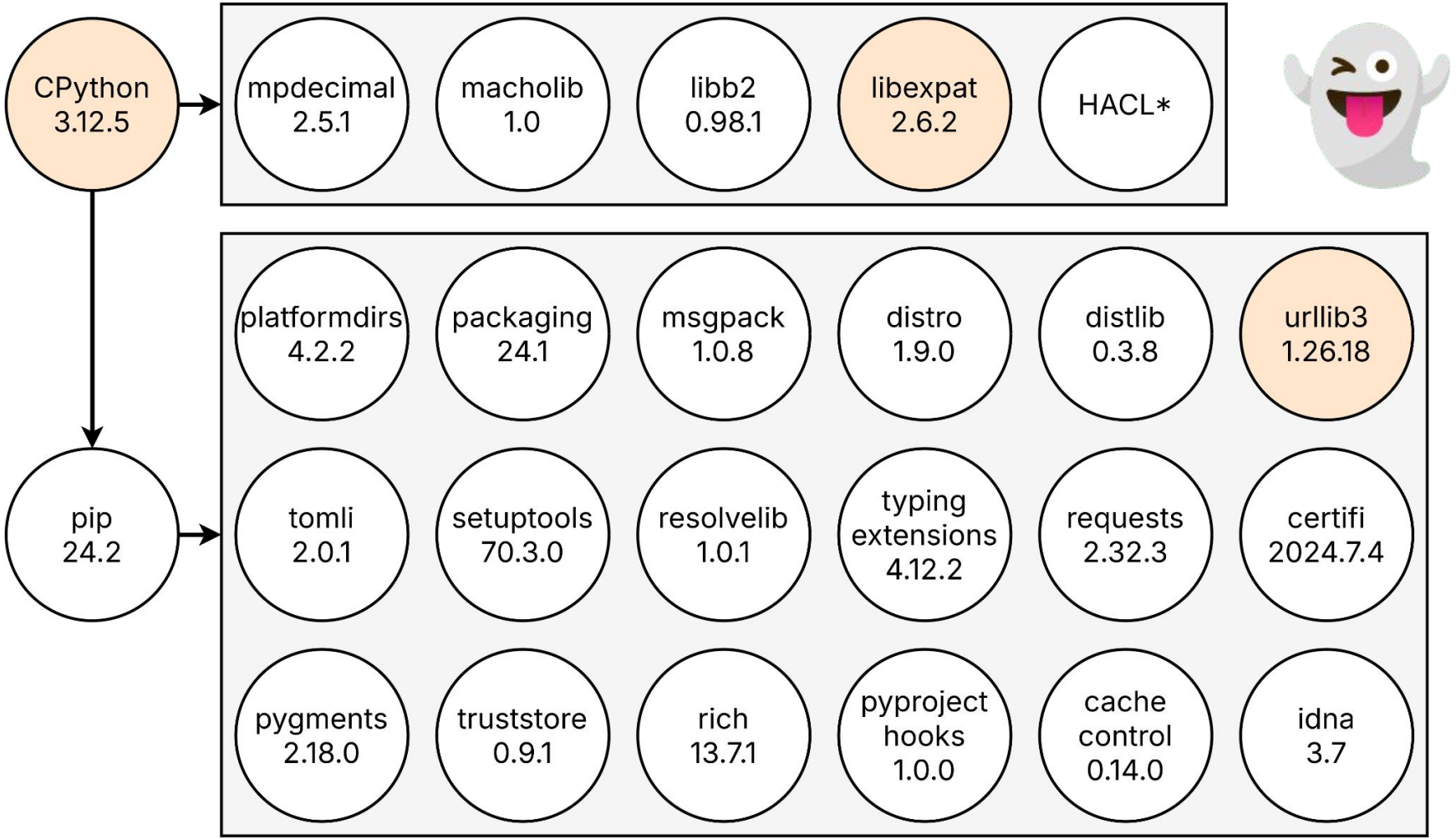
aqua trivy

$ python -m pip install pip-audit

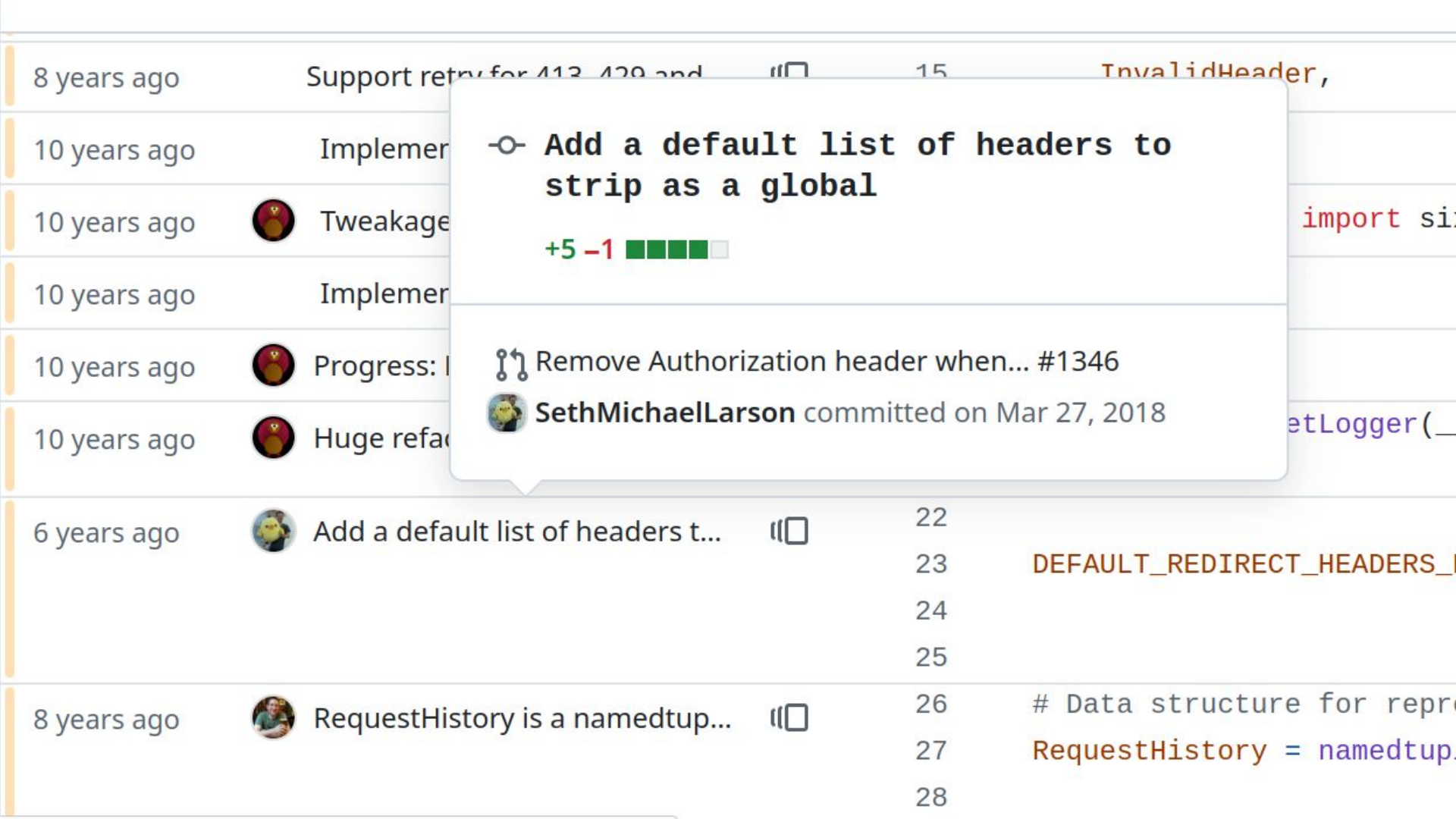
$ pip-audit requirements.txt

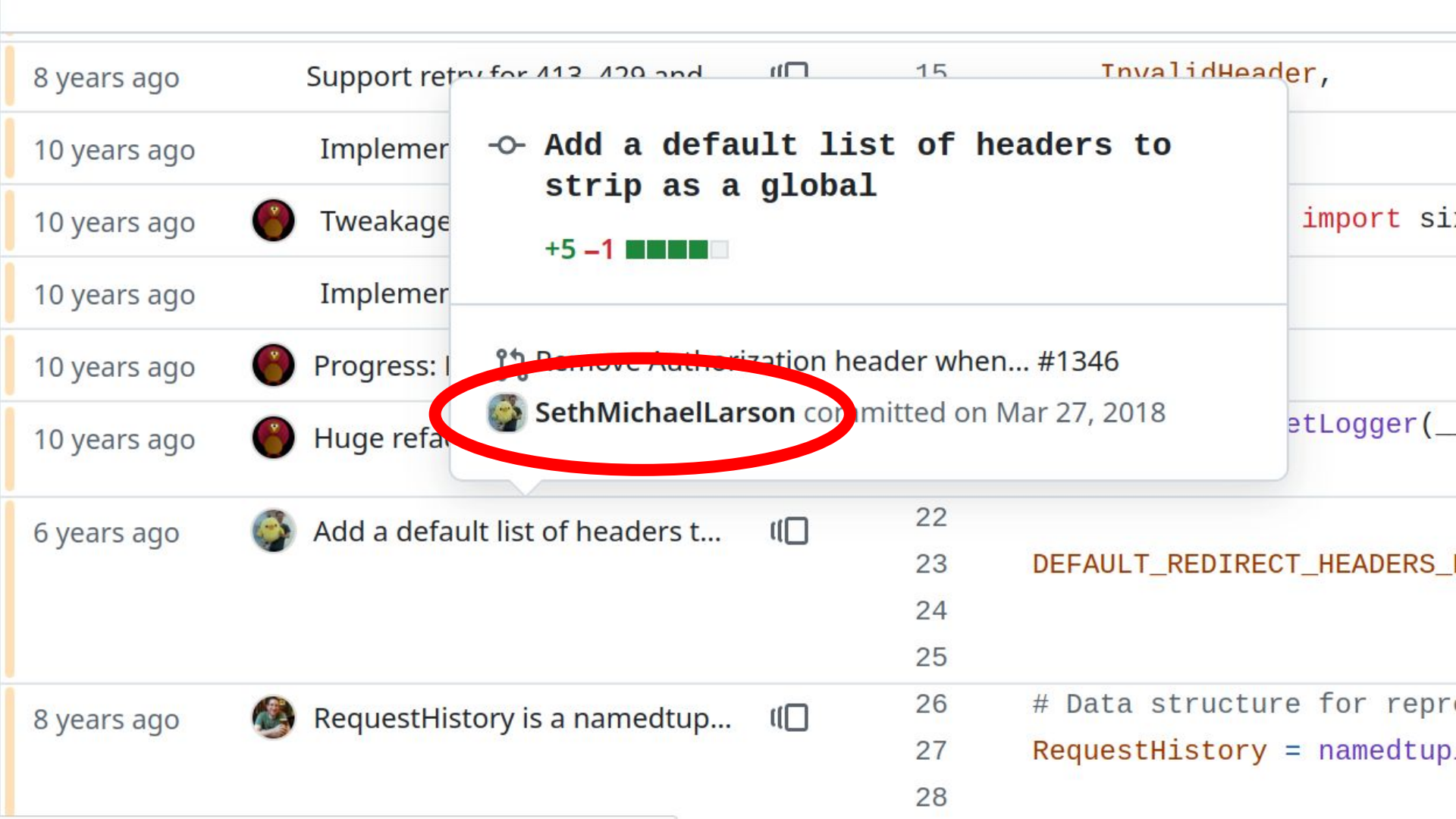
```
$ grype --by-cve sbom:Python-3.12.5.tgz.spdx.json

 ✔ Vulnerability DB                [updated]
 ✔ Scanned for vulnerabilities     [7 vulnerability matches]
   ├── by severity: 3 critical, 2 high, 1 medium, 0 low, 0 negligibl
   └── by status:   1 fixed, 6 not-fixed, 0 ignored

NAME     INSTALLED  FIXED-IN  TYPE    VULNERABILITY   SEVERITY
-------------------------------------------------------------------
CPython  3.12.5                       CVE-2024-7592   High
CPython  3.12.5                       CVE-2024-6232   High
CPython  3.12.5                       CVE-2024-8088   Unknown
expat    2.6.2                        CVE-2024-45492  Critical
expat    2.6.2                        CVE-2024-45491  Critical
expat    2.6.2                        CVE-2024-45490  Critical
urllib3  1.26.18    1.26.19   python  CVE-2024-37891  Medium
```

```
$ grype --by-cve sbom:Python-3.12.5.tgz.spdx.json

 ✔ Vulnerability DB                [updated]
 ✔ Scanned for vulnerabilities     [7 vulnerability matches]
   ├── by severity: 3 critical, 2 high, 1 medium, 0 low, 0 negligibl
   └── by status:   1 fixed, 6 not-fixed, 0 ignored

NAME     INSTALLED  FIXED-IN  TYPE    VULNERABILITY   SEVERITY
-------------------------------------------------------------------
CPython  3.12.5                       CVE-2024-7592   High
CPython  3.12.5                       CVE-2024-6232   High
CPython  3.12.5                       CVE-2024-8088   Unknown
expat    2.6.2                        CVE-2024-45492  Critical
expat    2.6.2                        CVE-2024-45491  Critical
expat    2.6.2                        CVE-2024-45490  Critical
urllib3  1.26.18    1.26.19   python  CVE-2024-37891  Medium
```


CPython
3.12.5
mpdecimal
2.5.1
macholib
1.0
libb2
0.98.1
libexpat
2.6.2
HACL*
pip
24.2
platformdirs
4.2.2
packaging
24.1
msgpack
1.0.8
distro
1.9.0
distlib
0.3.8
urllib3
1.26.18
tomli
2.0.1
setuptools
70.3.0
resolvelib
1.0.1
typing
extensions
4.12.2
requests
2.32.3
certifi
2024.7.4
pygments
2.18.0
truststore
0.9.1
rich
13.7.1
pyproject
hooks
1.0.0
cache
control
0.14.0
idna
3.7

```
$ grype --by-cve sbom:Python-3.12.5.tgz.spdx.json

 ✔ Vulnerability DB                [updated]
 ✔ Scanned for vulnerabilities     [7 vulnerability matches]
   ├── by severity: 3 critical, 2 high, 1 medium, 0 low, 0 negligibl
   └── by status:   1 fixed, 6 not-fixed, 0 ignored

NAME     INSTALLED  FIXED-IN  TYPE    VULNERABILITY   SEVERITY
-------------------------------------------------------------------
CPython  3.12.5                       CVE-2024-7592   High
CPython  3.12.5                       CVE-2024-6232   High
CPython  3.12.5                       CVE-2024-8088   Unknown
expat    2.6.2                        CVE-2024-45492  Critical
expat    2.6.2                        CVE-2024-45491  Critical
expat    2.6.2                        CVE-2024-45490  Critical
urllib3  1.26.18    1.26.19   python  CVE-2024-37891  Medium
```


8 years ago
Support retry for 413, 429 and
15 InvalidHeader,
10 years ago
Implemer
10 years ago
Tweakage
import si
10 years ago
Implemer
10 years ago
Progress:
10 years ago
Huge refa
etLogger(_
6 years ago
Add a default list of headers t...
8 years ago
RequestHistory is a namedtup...
Add a default list of headers to strip as a global
+5 −1
Remove Authorization header when... #1346
SethMichaelLarson committed on Mar 27, 2018
22
23 DEFAULT_REDIRECT_HEADERS_
24
25
26 # Data structure for repr
27 RequestHistory = namedtup
28


8 years ago
Support retry for 413, 429 and
10 years ago
Implemer
10 years ago
Tweakage
10 years ago
Implemer
10 years ago
Progress: I
10 years ago
Huge refa
6 years ago
Add a default list of headers t...
8 years ago
RequestHistory is a namedtup...
Add a default list of headers to strip as a global
+5 −1
Remove Authorization header when... #1346
SethMichaelLarson committed on Mar 27, 2018
15 InvalidHeader,
import si
etLogger(__
22
23 DEFAULT_REDIRECT_HEADERS_
24
25
26 # Data structure for repr
27 RequestHistory = namedtup
28

No shame in CVEs 😅

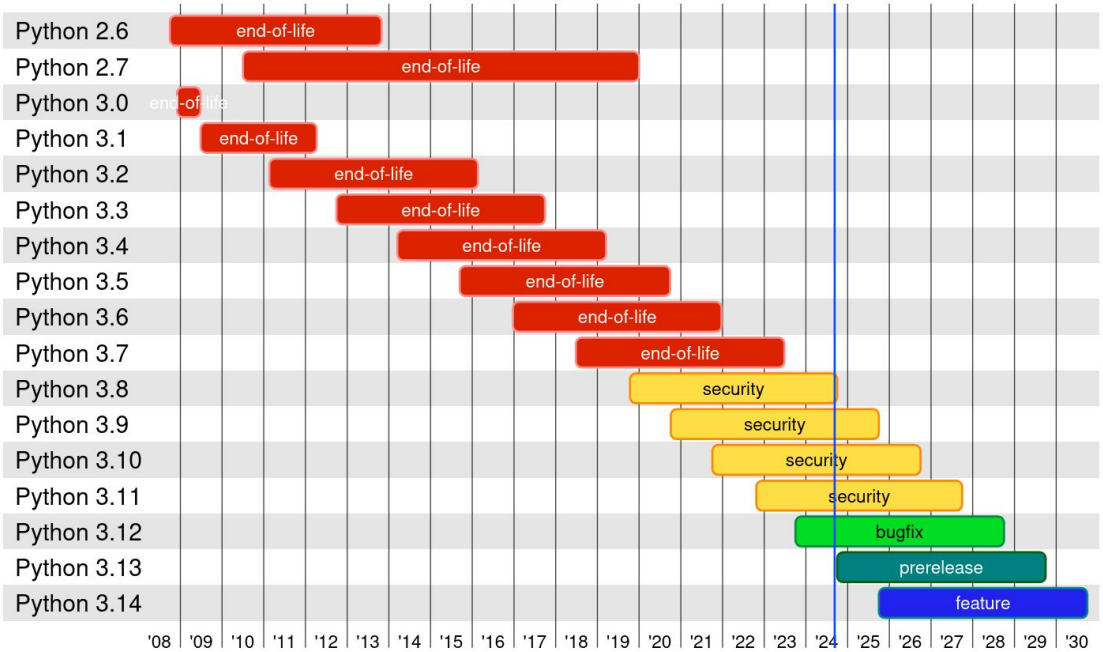
CVEs can be a ***positive*** indicator of security!

(zero fixed CVEs might mean new or untested)

Maintainers: Don’t stress over the process.

Fix the issue, make the fix available, publish CVE

Are you getting security fixes?



https://devguide.python.org/versions

CPython provides 5 years of security fixes

Some distributors backport fixes to older versions

↑ High demand for SBOMs

→ SBOM strategy for Python packages

Author standards (PEPs) and enable SBOM tools
Some work in progress or done (PEP 639, PEP 710)

Minimize burden on open source maintainers

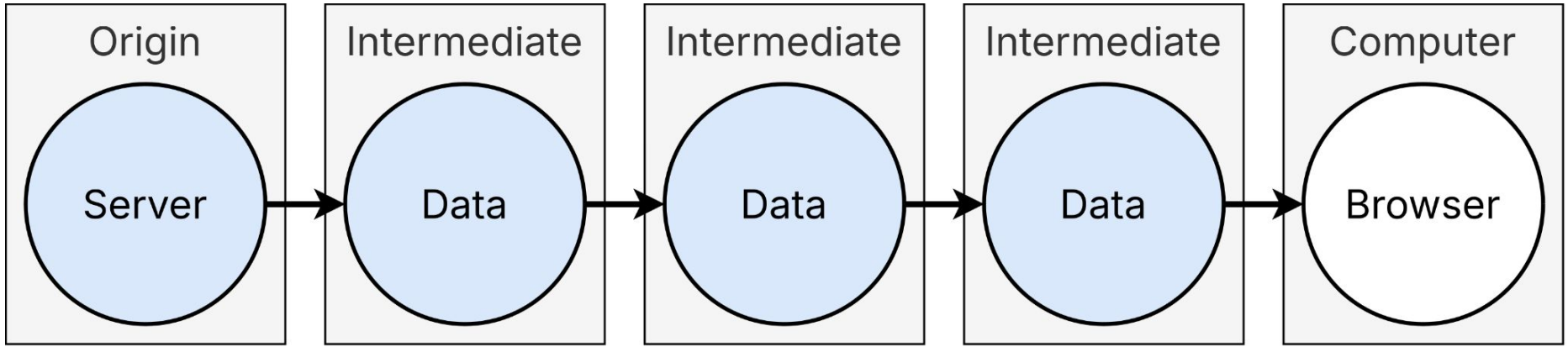
Building a security culture:

- Maintainers: Adopt lightweight security policy
- Users: Add a vulnerability scanner
- Users: Fixing CVEs: usually upgrades
- Users: Need an SBOM? Don’t ask, generate!

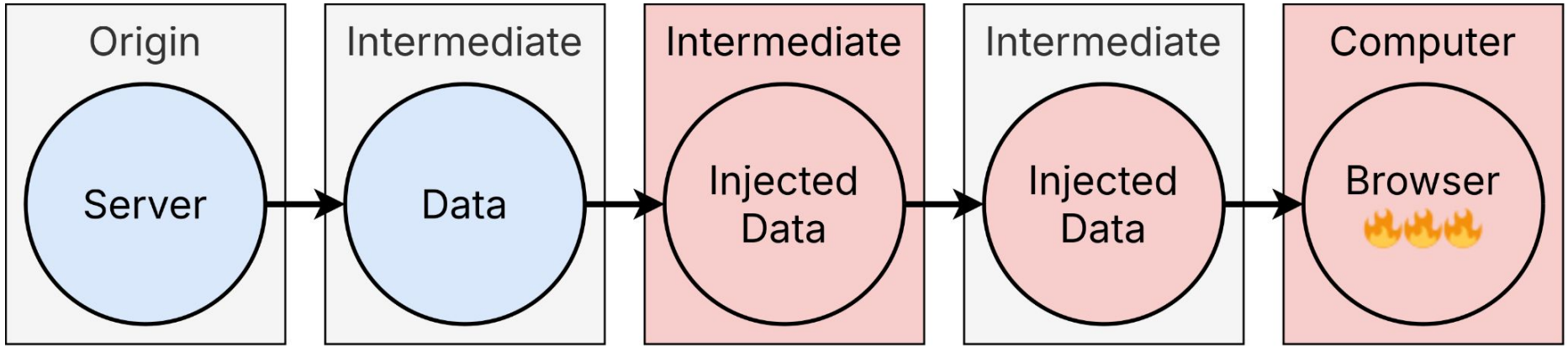
Chapter II: Pipes

Software in motion

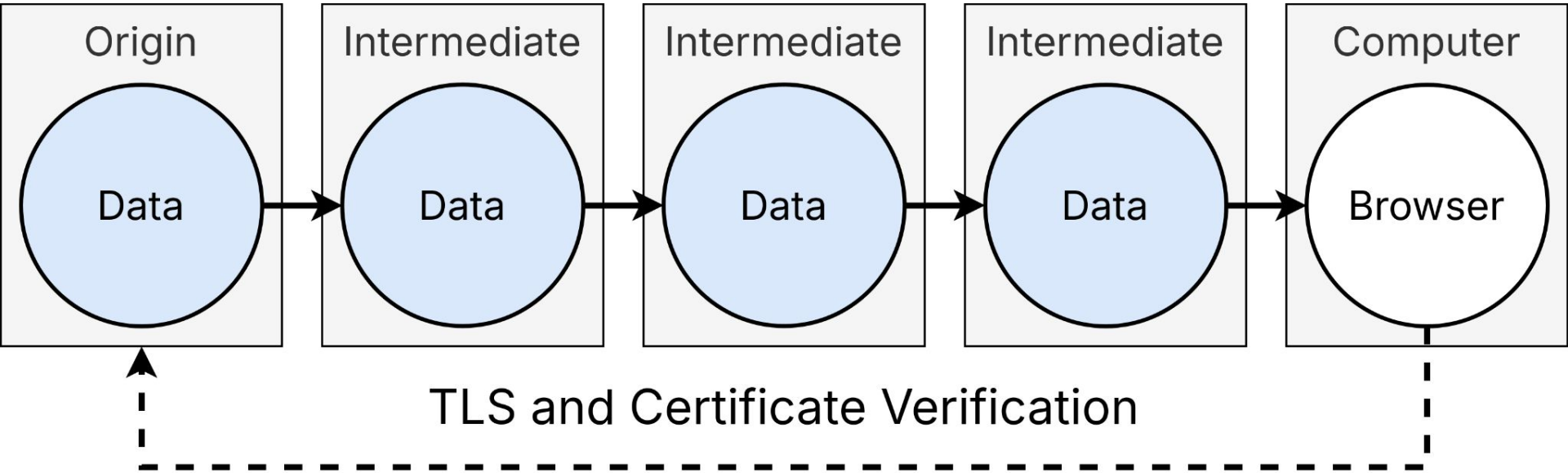
Internet → Network of machines



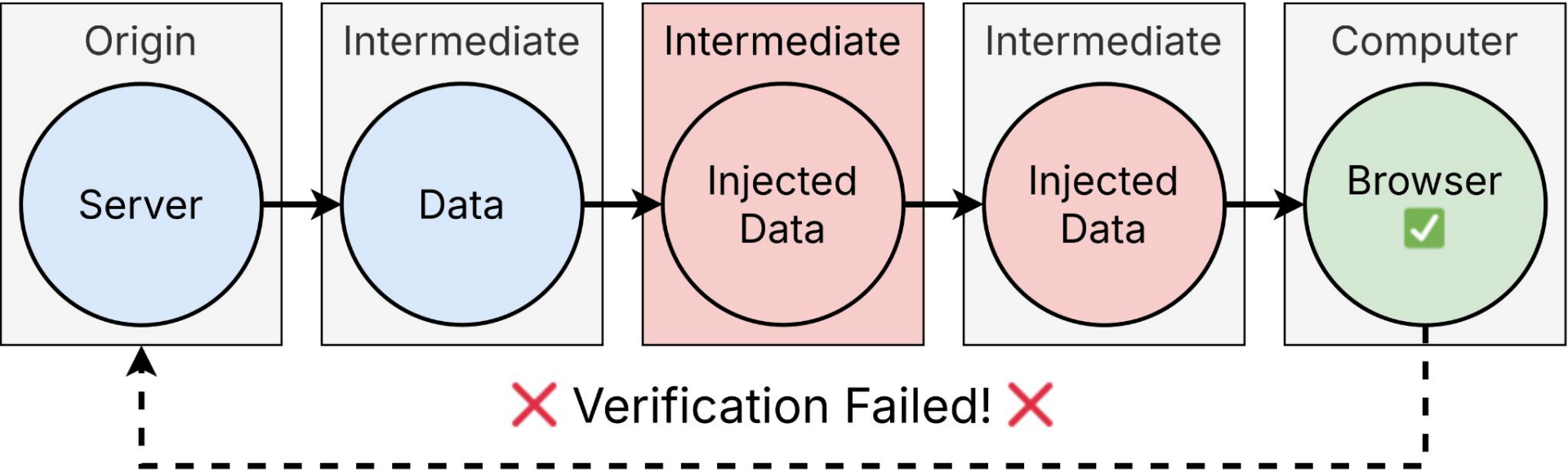
Internet → Network of machines



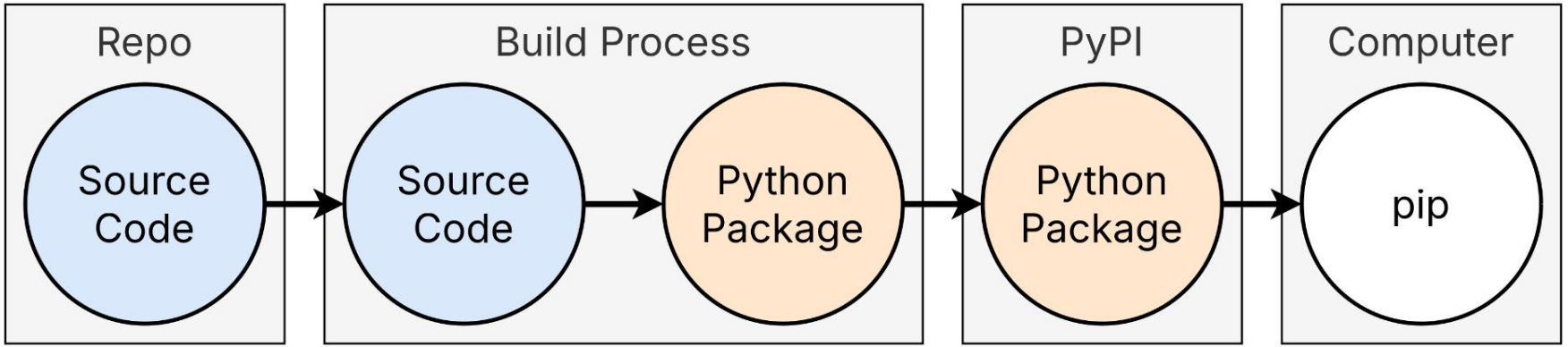
HTTPS: Untrusted → Trusted



HTTPS: Untrusted → Trusted



Can we do this for software?



“Securing software in motion”

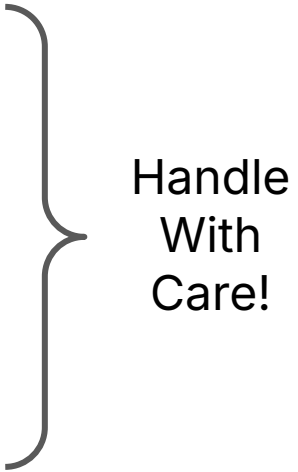
Learn about threats? https://slsa.dev/spec/v1.0/threats

Fragile: Handle with Care

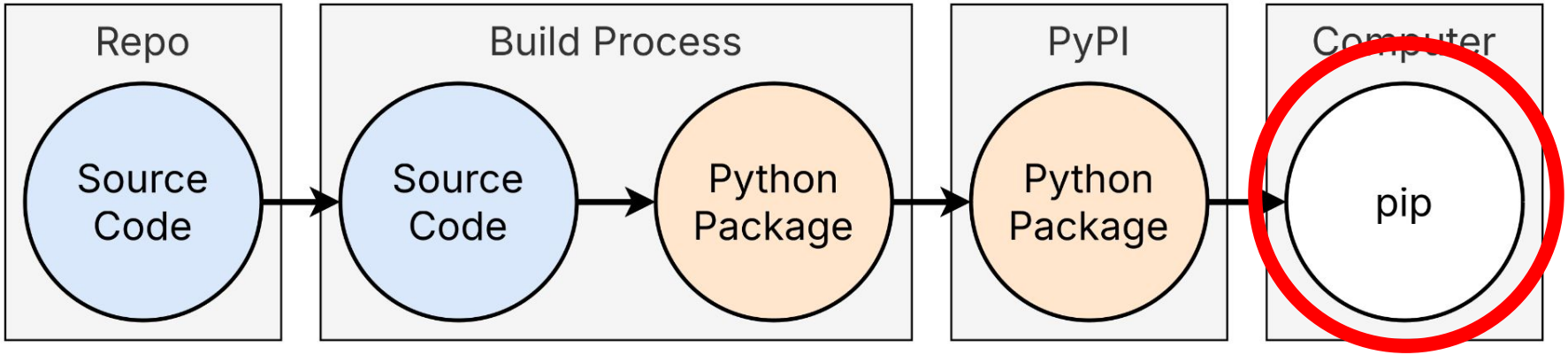
Software at rest: Limited to existing vulnerabilities

Software in motion: ***Opportunity* to inject bad code**

- Adding or installing new dependencies
- Deploying software to another machine
- Getting data or software from the internet



Adding new dependencies



Adding new dependencies

Don’t use **unverified data!** Attackers fake these:

- Number of Stars, Favorites, Downloads
- Links to source, documentation, websites
- Using names of trusted entities (“django”, “python”)
- Names similar to known projects (“requestz”)

Don’t trust random strangers on the internet:

→ *pip install means running code!*

Adding new dependencies

Make choices using ***verifiable data!*** ✅ Ask:

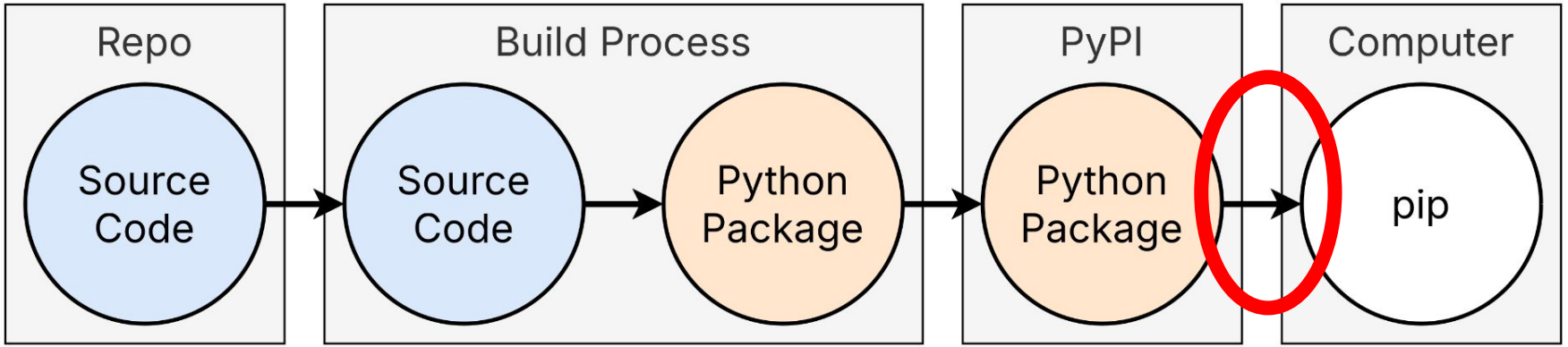
- Do we already use this dependency elsewhere?
- Do the maintainer(s) also maintain other projects?
- How many projects depend on this project?
- How long has a project been published?
- What do other people I trust think about this project?

Adding new dependencies

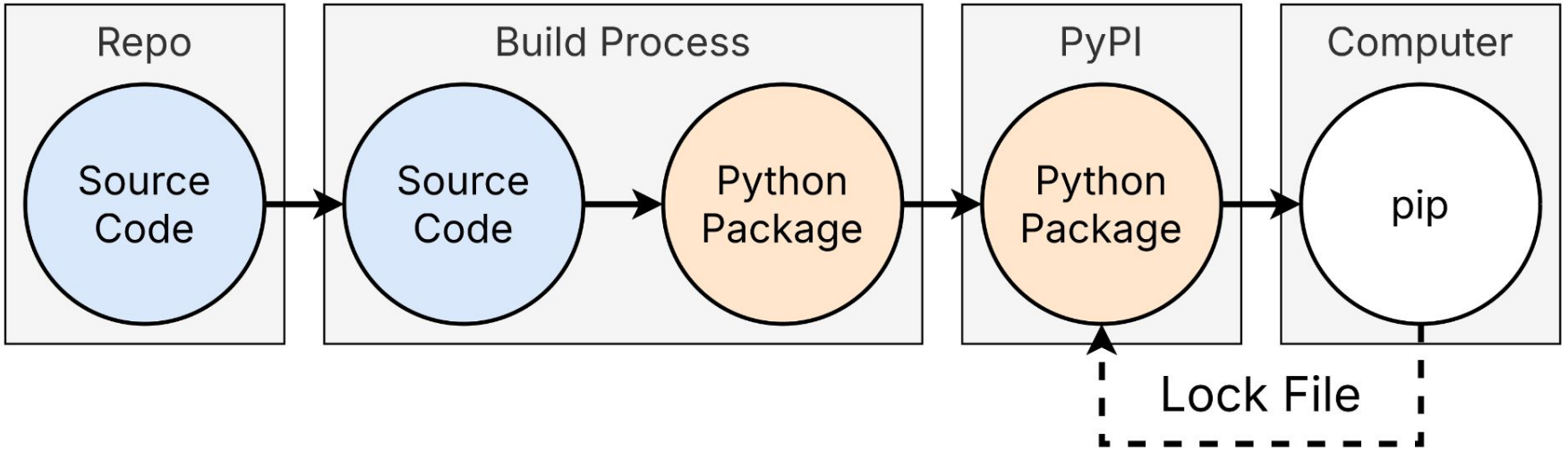
Where to get free data on open source projects?

- Ecosyste.ms: https://packages.ecosyste.ms
- Libraries.io: https://libraries.io
- Trusty: https://trustypkg.dev
- Deps.dev: https://deps.dev

“Locking” dependencies



“Locking” dependencies



“Locking” dependencies

Why? Consistent dependencies across all environments

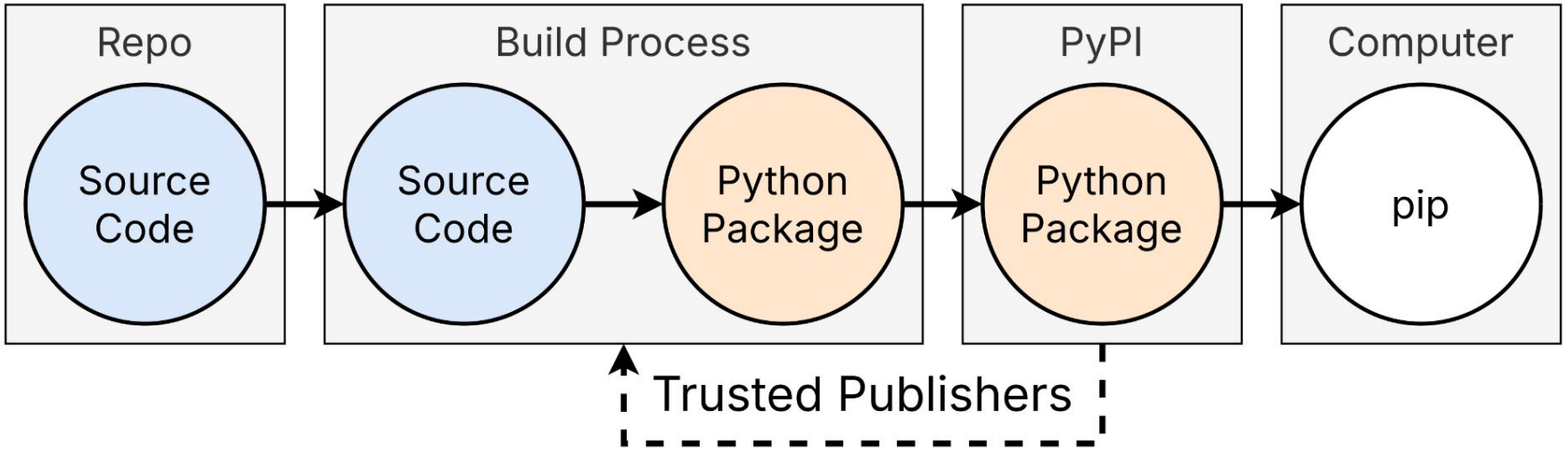
Good for *development and security* (Win-Win!)

How? pip-tools*, Poetry, PDM, uv

* Use --generate-hashes with pip-tools

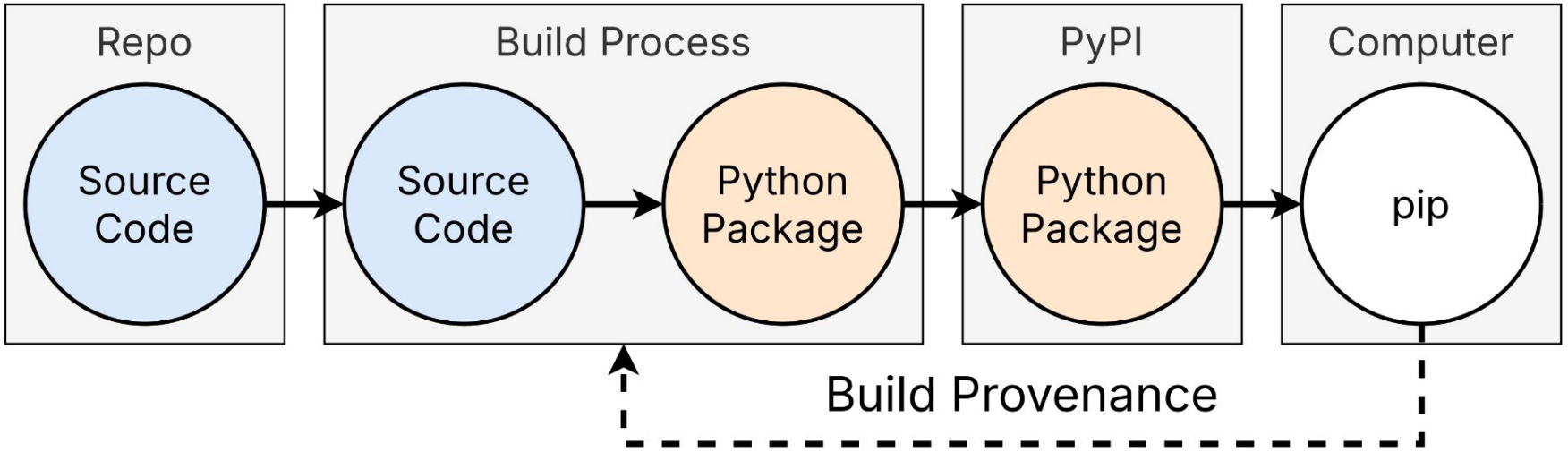
When? Applications, build and release

Trusted Publishers



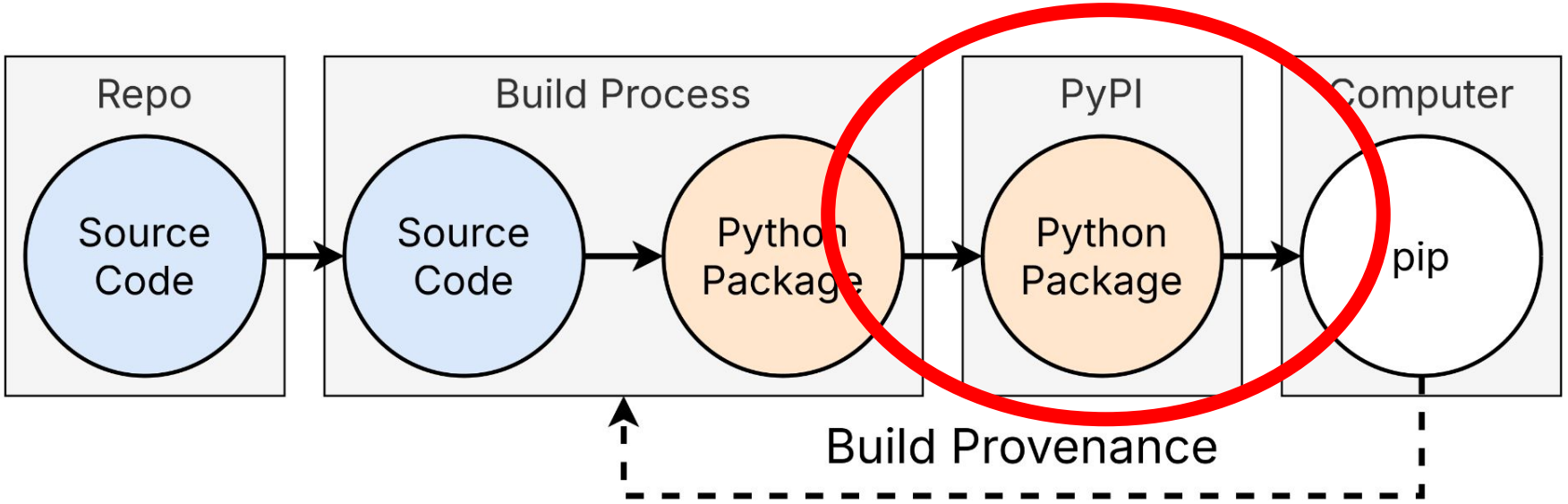
Publish to PyPI without a password or API key!

Provenance (“Origin”)



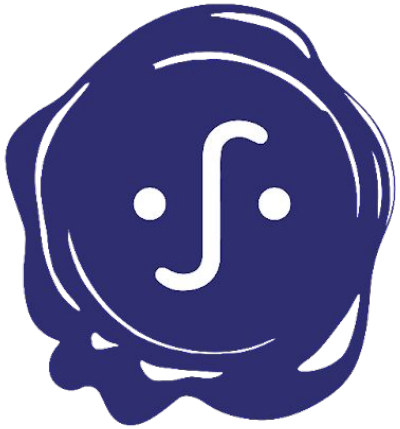
“HTTPS” for software: Prove where software came from end-to-end ***without trusting the middle!***

Provenance (“Origin”)



“HTTPS” for software: Prove where software came from end-to-end ***without trusting the middle!***

Build Provenance for CPython in Action!



sigstore

Files

Version	Operating System	Description	MD5 Sum	File Size	GPG	Sigstore	SBOM
Gzipped source tarball	Source release		d23d56b51d36a9d51b2b13d30c849d00	25.7 MB	SIG	.sigstore	SPDX
XZ compressed source tarball	Source release		02c7d269e077f4034963bba6befdc715	19.5 MB	SIG	.sigstore	SPDX
macOS 64-bit universal2 installer	macOS	for macOS 10.9 and later	78bd8d0795062b1df63e2b8d8386a5fa	43.5 MB	SIG	.sigstore	
Windows installer (64-bit)	Windows	Recommended	bbcb2fcf9d739f776fb6414afc12c80d	25.3 MB	SIG	.sigstore	SPDX
Windows installer (32-bit)	Windows		d151f5f116e11c4d40021527f51ddf67	24.0 MB	SIG	.sigstore	SPDX
Windows installer (ARM64)	Windows	Experimental	365d59eff83dfea9af528df4ebd060cb	24.5 MB	SIG	.sigstore	SPDX
Windows embeddable package (64-bit)	Windows		0f53697bdcecfb97b99ac8aa9d9a9e13	10.5 MB	SIG	.sigstore	SPDX
Windows embeddable package (32-bit)	Windows		82dd15b14c307f5fcef80ccb45d6b404	9.4 MB	SIG	.sigstore	SPDX
Windows embeddable package (ARM64)	Windows		62c81364c232644f280b06ef5f33a029	9.8 MB	SIG	.sigstore	SPDX

https://python.org/downloads

Files

Version	Operating System	Description	MD5 Sum	File Size	GPG	Sigstore	SBOM
Gzipped source tarball	Source release		d23d56b51d36a9d51b2b13d30c849d00	25.7 MB	SI	.sigstore	SPDX
XZ compressed source tarball	Source release		02c7d269e077f4034963bba6befdc715	19.5 MB	SIG		SPDX
macOS 64-bit universal2 installer	macOS	for macOS 10.9 and later	78bd8d0795062b1df63e2b8d8386a5fa	43.5 MB	SIG	.sigstore	
Windows installer (64-bit)	Windows	Recommended	bbcb2fcf9d739f776fb6414afc12c80d	25.3 MB	SIG	.sigstore	SPDX
Windows installer (32-bit)	Windows		d151f5f116e11c4d40021527f51ddf67	24.0 MB	SIG	.sigstore	SPDX
Windows installer (ARM64)	Windows	Experimental	365d59eff83dfea9af528df4ebd060cb	24.5 MB	SIG	.sigstore	SPDX
Windows embeddable package (64-bit)	Windows		0f53697bdcecfb97b99ac8aa9d9a9e13	10.5 MB	SIG	.sigstore	SPDX
Windows embeddable package (32-bit)	Windows		82dd15b14c307f5fcef80ccb45d6b404	9.4 MB	SIG	.sigstore	SPDX
Windows embeddable package (ARM64)	Windows		62c81364c232644f280b06ef5f33a029	9.8 MB	SIG	.sigstore	SPDX

https://python.org/downloads

```
$ python -m pip install sigstore

$ sigstore verify identity \
    --cert-identity thomas@python.org \
    --cert-oidc-issuer https://accounts.google.com \
    --bundle Python-3.12.5.tgz.sigstore \
    Python-3.12.5.tgz

OK: Python-3.12.5.tgz
```

```
$ python -m pip install sigstore

$ sigstore verify identity \
    --cert-identity thomas@python.org \
    --cert-oidc-issuer https://accounts.google.com \
    --bundle Python-3.12.5.tgz.sigstore \
    Python-3.12.5.tgz

OK: Python-3.12.5.tgz
```

```
$ python -m pip install sigstore

$ sigstore verify identity \
    --cert-identity thomas@python.org \
    --cert-oidc-issuer https://accounts.google.com \
    --bundle Python-3.12.5.tgz.sigstore \
    Python-3.12.5.tgz

OK: Python-3.12.5.tgz
```

File is from Thomas!

PyPI: New features coming!

https://blog.pypi.org

Building a security culture:

- Maintainers: Trusted Publishers!
- Users: Create policy for using open source
- Everyone: Use lock files for dependencies
- Everyone: Follow the PyPI blog for news

Chapter 3: People

Software as a Story

```
$ python -m pip install <package>
```

$ python -m pip install <package> 💥

SSLCertVerificationError(
1, '[SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed: unable to get local issuer certificate (_ssl.c:1045))

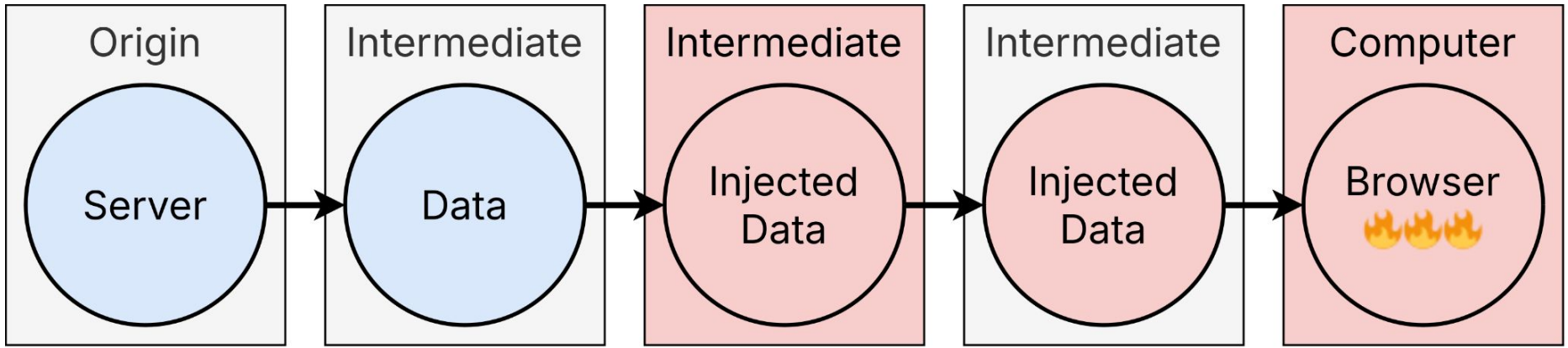
→ How would you solve this problem?
StackOverflow? ChatGPT?

Debugging...

Question: pip install, receiving error, what to do?

StackOverflow (10 years ago, 1K upvote, 2.3 million views)

- Use --trusted-host 🔥


Origin
Server
Intermediate
Data
Intermediate
Injected Data
Intermediate
Injected Data
Computer
Browser

Debugging...

Question: pip install, receiving error, what to do?

StackOverflow (10 years ago, 1K upvote, **2.3 million views**)

- Use --trusted-host 🔥
- Add trusted-host to pip.conf 🔥🔥🔥

How many insecure configurations are out there? 😨

Debugging...

Question: pip install, receiving error, what to do?

ChatGPT (GPT-4):

- Manually install certificates
- Upgrade certifi
- Upgrade pip
- Use --trusted-host 🔥

Debugging...

Question: pip install, receiving error, what to do?

ChatGPT (GPT-4):

- Manually install certificates
- Upgrade certifi
- Upgrade pip ← **Correct answer, but...**
- Use --trusted-host 🔥

Seth Michael Larson
@sethmlarson

I've been working on a potential replacement to #Python certifi called "truststore" which uses native operating system trust anchors on macOS and Windows to verify certificates 🎉

Drop-in SSLContext works with @urllib3 and aiohttp. Currently macOS is working and Windows is next:

```python
import urllib3
import truststore  # New experimental library!

# SSLContext that uses system certificates
ctx = truststore.TruststoreSSLContext()
```

197 ❤ 33 🔁

March 2022...

pip: August 2024

1.5 years, why...?

No feedback from users...

No breaking!

Paying tech debt at the ecosystem-scale 😓

- **Adding security *after* is harder... (but we are here)**
- Breaking changes, hard to be confident...
- Old information is popular, how to help users?
- User feedback or pre-releases is rare. No telemetry

Find and fix insecure code!

Bandit

$ python -m pip install bandit

$ bandit

Maintainers: Contribute detections!

Testing with warnings

Warnings not shown by default: **must enable!**

Early notice for when software is changing

PYTHONWARNINGS=error::DeprecationWarning

Start conversations!

Maintainers: Put URLs in warnings and errors.
Send users to documentation or issue tracker

Users: keep maintainers informed of new issues.
Find documentation or issue tracker,
third-party info can be outdated

💥 Breaking changes 💥

Last resort, but may be needed!

Users: Accept them with grace, be collaborative:
Describe any use-cases that aren’t covered

Maintainers: Document and warn as soon as possible
Yanking to “roll-back” releases if needed

Securing your local Python community:

What can you do to help?

- Learn, share, and adapt stories about security in the communities you care about.

- Take action! Adopt one or more practices, set an example, update documentation

Securing your local Python community:

What can you do to help?

- Create a security policy
- Adopt Trusted Publishers, secure release process
- Deprecate and remove insecure defaults
- Apply for funding for larger security projects

Securing your local Python community:

What can you do to help?

SPEC 8: Securing the Release Process

Permissions, pinned workflows,
Trusted Publishers, and SLSA ⭐

https://scientific-python.org/specs/spec-0008

Users of open source Python packages:

What can you do to help?

Create policy for ***using and contributing***
to open source projects you depend on

Adopt one or more security tools
and take action from the results


401
EXIT
Barry
Hugo
Tian
Carol
Irit
Python Language Summit 2024

Let's secure Python, *together!*