Software Heritage for Open Source Compliance

Stefano Zacchiroli

Polytechnic Institute of Paris & Software Heritage
stefano.zacchiroli@telecom-paris.fr
@zacchiro@mastodon.xyz

25 November 2025

Software Heritage
THE GREAT LIBRARY OF SOURCE CODE

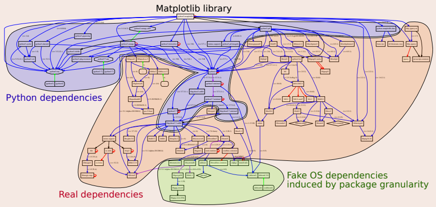
Outline

About the speaker

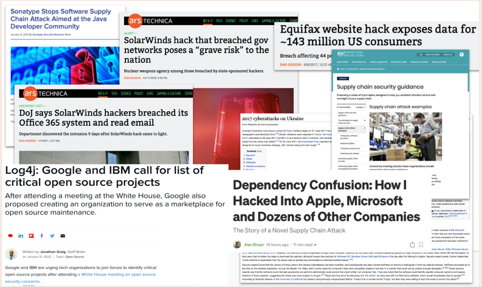
- Professor of Computer Science, Télécom Paris, Polytechnic Institute of Paris
- Free/Open Source Software activist (25+ years)
- Debian Developer & Former 3x Debian Project Leader
- Former Open Source Initiative (OSI) director
- Software Heritage co-founder & CSO
- SECUBIC site lead for Télécom Paris

Enhancing software Reuse, Security and Transparency

Software complexity is growing…



…you need to Know Your Software (KYSW)



Sonatype Stops Software Supply Chain Attack Aimed at the Java Developer Community

SolarWinds hack that breached gov networks poses a "grave risk" to the nation

Nuclear weapons agency among those breached by state-sponsored hackers.

Equifax website hack exposes data for ~143 million US consumers

Supply chain security guidance

Supply chain attack examples

DoJ says SolarWinds hackers breached its Office 365 system and read email

Department discovered the intrusion 9 days after SolarWinds hack came to light.

2017 cyberattacks on Ukraine

Log4j: Google and IBM call for list of critical open source projects

After attending a meeting at the White House, Google also proposed creating an organization to serve as a marketplace for open source maintenance.

Google and IBM are urging tech organizations to join forces to identify critical open source projects after attending a White House meeting on open source security concerns.

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

The Story of a Novel Supply Chain Attack

Sec. 4. Enhancing Software Supply Chain Security

ensuring and attesting […] to the integrity and provenance of open source software

May 2021 POTUS Executive Order

EU Cyber Resilience Act (2024/2847)

Regulation aims to […] ensuring […] software products […] with fewer vulnerabilities.

We need a *trusted* knowledge base providing *software integrity and provenance*!

Outline

Software Heritage, in a nutshell

Software Heritage
THE GREAT LIBRARY OF SOURCE CODE

Collect, preserve and share *all* software source code

Preserving our heritage, enabling better software and better science for all

Reference catalog

find and reference all software source code

Universal archive

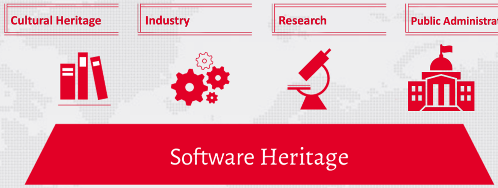
preserve and share all software source code

Research infrastructure

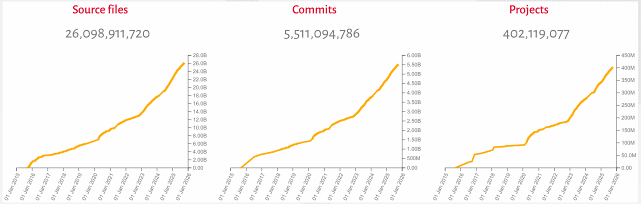
enable analysis of all software source code

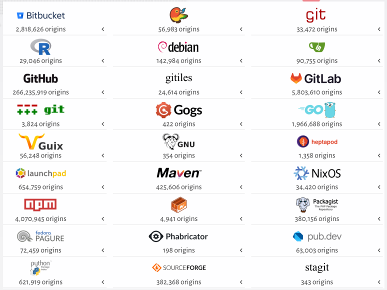
The largest software archive, a shared infrastructure

One infrastructure
open and shared



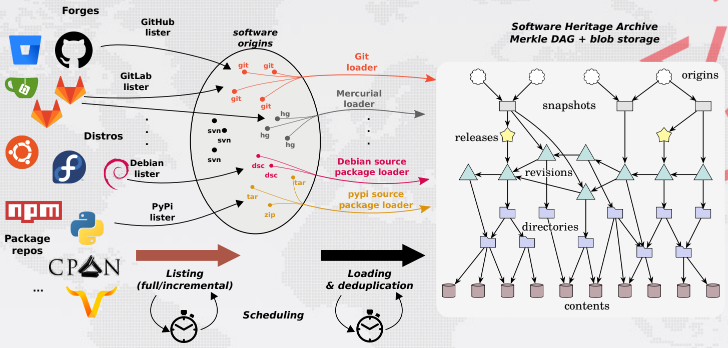
The largest archive ever built





Bitbucket 2,818,626 origins	56,983 origins	git 33,472 origins
R 29,046 origins	debian 142,984 origins	90,755 origins
GitHub 266,235,919 origins	gitiles 24,614 origins	GitLab 5,803,610 origins
git 3,824 origins	Gogs 422 origins	GO 1,966,688 origins
Guix 56,248 origins	GNU 354 origins	heptapod 1,358 origins
launchpad 654,759 origins	Maven 425,606 origins	NixOS 34,420 origins
npm 4,070,945 origins	4,941 origins	Packagist The PHP Package Repository 380,156 origins
fedora PAGURE 72,459 origins	Phabricator 198 origins	pub.dev 63,003 origins
python Package Index 621,919 origins	SOURCEFORGE 382,368 origins	stagit 343 origins

A peek under the hood: a universal archive

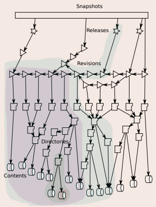


Global development history permanently archived in a uniform data model

- over 24 billion unique source files from over 375 million software projects
- ~2PB (compressed) blobs, ~50 B nodes, ~900 B edges

A revolutionary infrastructure for industry

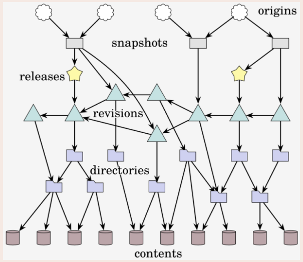
The *graph* of public software development



All of the software development in a single graph!

- lookup by content hash
- wayback machine for software development
  - http://archive.softwareheritage.org/
- … and much more

The *global ledger* of public code

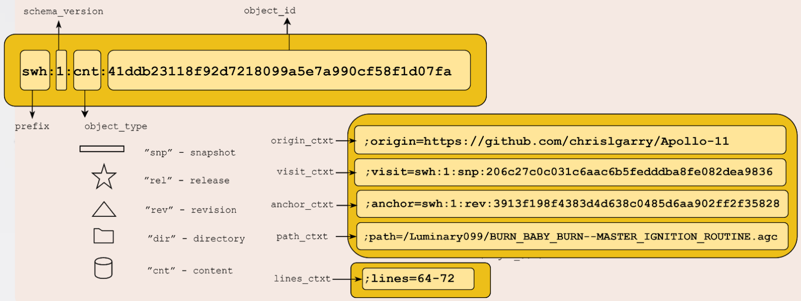


All of a software development… in a single Merkle graph!
Widely used crypto (e.g., Git, blockchains, IPFS, …)

- built-in deduplication
- intrinsic, unforgeable identifiers at all levels
- simplifies traceability (licensing, supply chain management)

Referencing all source code artifacts with SWHIDs

Software Heritage Identifiers (SWHID) see swhid.org



schema_version
object_id
swh:1:cnt:41ddb23118f92d7218099a5e7a990cf58f1d07fa
prefix
object_type
"snp" - snapshot
"rel" - release
"rev" - revision
"dir" - directory
"cnt" - content
origin_ctxt
visit_ctxt
anchor_ctxt
path_ctxt
lines_ctxt
;origin=https://github.com/chrislgarry/Apollo-11
;visit=swh:1:snp:206c27c0c031c6aac6b5fedddba8fe082dea9836
;anchor=swh:1:rev:3913f198f4383d4d638c0485d6aa902ff2f35828
;path=/Luminary099/BURN_BABY_BURN--MASTER_IGNITION_ROUTINE.agc
;lines=64-72

50+B
intrinsic,
decentralised,
cryptographic

Full fledged *source code references* for traceability, integrity and reproducibility

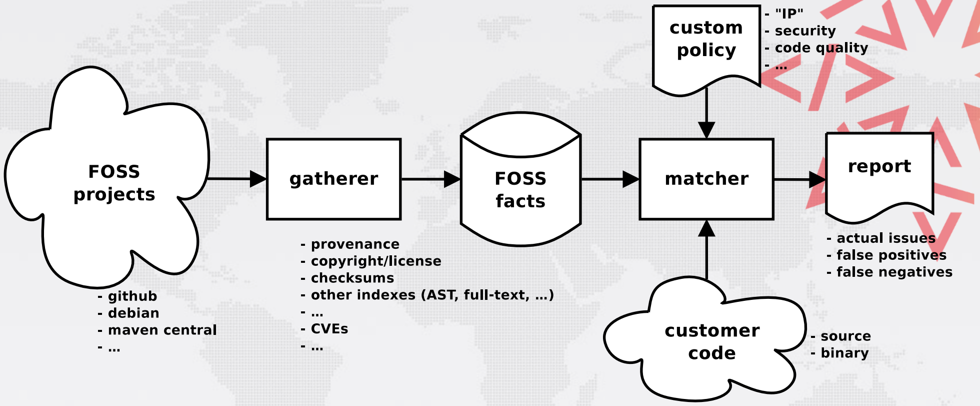
- Linux Foundation SPDX 2.2
- IANA-registered "swh:"
- WikiData property P6138

Examples: Apollo 11 AGC excerpt, Quake III rsqrt
Guidelines available, see the HOWTO

ISO/IEC 18670, see swhid.org

Outline

Anatomy of a KYSW toolchain



A code scanner is the key ingredient of all KYSW toolchains: it scans a local source code base and compares it to a FOSS knowledge base, summarizing findings.

SWH Scanner

Vision

swh-scanner is an open source and open data source code scanner for open compliance workflows, backed by the largest public archive of FOSS source code.

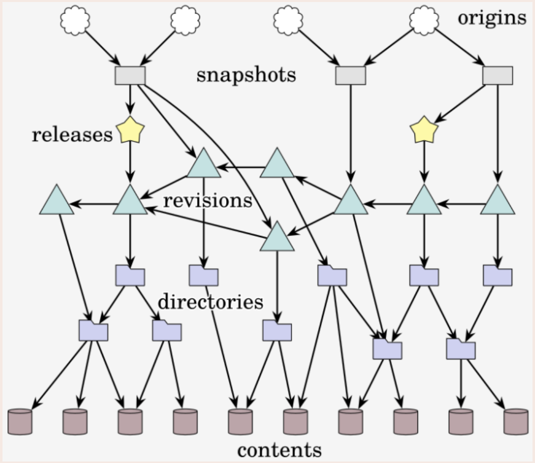
Design

- Partition a source tree into known (= published before) v. unknown
- Provide provenance information on demand
- Software Heritage Archive as ground truth for public code
- Merkle DAG model and SWHIDs for maximum efficiency
- File-level granularity

Code: gitlab.softwareheritage.org/swh/devel/swh-scanner (GPL 3+)
Package: pypi.org/project/swh.scanner

Leveraging the Software Heritage data model for efficient scanning

Merkle DAG



Efficient scanning

- If a node (e.g., the root directory of a project) is known to the Software Heritage archive, all contained files and directories are known as well → no need to query for them!
- If a node is not known, we recurse to children and stop querying when reaching known nodes (e.g., embedded copies of 3rd party FOSS code or previous versions)

Daniele Serafini, Stefano Zacchiroli
Efficient Prior Publication Identification for Open Source Code
OSS+OpenSym 2022. ACM 2022.
https://hal.science/hal-03735961/

Demo

Setup

```
$ pip install swh-scanner

$ swh scanner setup
$ swh scanner scan $PROJECT_PATH
```

Demo

swh-scanner demo — Efficiency

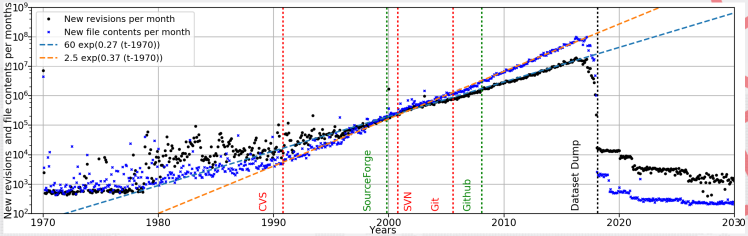
```
$ du -sh --exclude=.git /srv/src/linux/git
4,1G /srv/src/linux/git
```

```
$ time swh scanner scan /srv/src/linux
Files:                  78277
            known:      78267 ( 99%)
directories:             5085
      fully-known:       5081 ( 99%)
  partially-known:          4 (  0%)

38,65s user 4,71s system 81% cpu 53,127 total
```

```
$ swh scanner scan --output-format ndjson /srv/src/linux/git | grep false
...
{"scripts/kconfig/symbol.o": {"swhid": "swh:1:cnt:874f19...", "known": false}}
...
```

Source code provenance



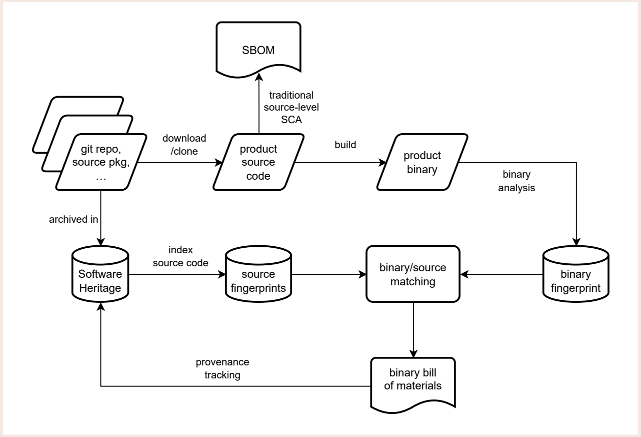
Key findings

- The amount of original commits in public code doubles every ~30 months and has been doing so for 20+ years; original source code files double every ~22 months
- It is possible to trace the provenance of source code artifacts at this scale in a compact relational model via the notion of isochrone graphs.

Rousseau, Di Cosmo, Zacchiroli
Software Provenance Tracking at the Scale of Public Source Code
Empir. Softw. Eng. 25(4): 2930-2959 (2020)

Outline

SWH @ SECUBIC — Vision



- Leverage Software Heritage as a comprehensive archive of public code, to extract source code fingerprints that can be matched to binaries, e.g., symbols (for unstripped binaries), static data, etc.
- Extract binary fingerprints from binaries of interest; match them against the source fingerprints database
- Use Software Heritage provenance information to identify origin

SWH @ SECUBIC — Challenges

Scale

- Indexing the 2 PiB of source code archived by Software Heritage is not feasible, and out of scope for SECUBIC
- The focus of SECUBIS is on developing scalable techniques, not creating the actual fingerprints base
  - Industrial partners might be interested in picking this up though!
- Approach: focus on subset of interests of the archive, e.g., by programming language ecosystem [a]

[a] Sun, German, Zacchiroli. Using the uniqueness of global identifiers to determine the provenance of Python software source code. Empir Software Eng 28, 107 (2023).

Ambiguity

- How many open source projects contain a `print` function?
- How many versions of the same project do?

Appendix

Open compliance vs Source code scanning

Definition (Open Compliance)

The pursuit of compliance with *license obligations* and other *best practices* for the management of open source software components, using only open technologies such as: open source software, open data information, and open access documentation.

Why

Reduced lock-in risks, lower total cost of ownership (TCO), crowdsourcing, alignment with FOSS community ethos.

Q: Can we build an industry-grade source code scanning tool, compliant with Open Compliance principles, on top of Software Heritage?