Software Heritage

An Unique Infrastructure for Big Code Research

Stefano Zacchiroli

Télécom Paris, Institut Polytechnique de Paris & Software Heritage
stefano.zacchiroli@telecom-paris.fr

4 November 2026 — Inria, La Revue Scientifique



Software Heritage

THE GREAT LIBRARY OF SOURCE CODE

Software science is blind without data



Photo: ALMA (ESO/NAOJ/NRAO), R. Hills

A wealth of research on software rests on the code itself

- Software engineering and evolution
- Cybersecurity and the supply chain
- AI on code: training, evaluation, provenance
- Innovation economics: who builds software, where; what's its impact
- Software architecture, verification, …

Common need: observe all the code, not a sample of it: not just GitHub, not just today.

If you study the stars, you go to Atacama Desert for the VLT.
Where is the Very Large Telescope of code?

Software Heritage: the very large telescope of source code

One infrastructure
open and shared

Started by Inria in
2016, with UNESCO



The largest public archive of source code ever built: 30B unique files, 6B commits, 450M projects





What makes it unique for research

- Universal: GitHub, GitLab, university forges, Linux distributions, package managers, rescued legacy code

  no single-platform bias

- Uniform: git, svn, hg, tarballs, … *one data model*

  VCS and forge details abstracted away

- Deduplicated: every file, directory, commit stored *once*, across all projects

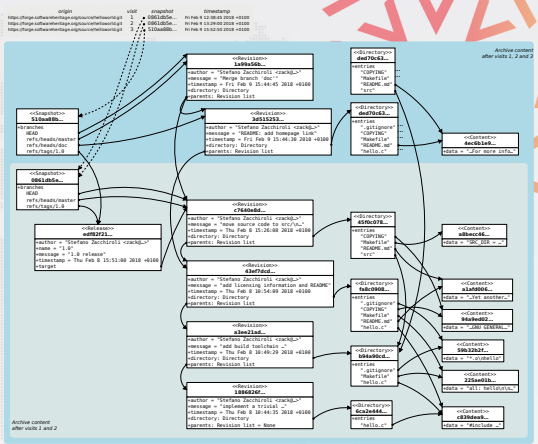
  "where else does this code live?" becomes answerable

- Historical: 50+ years of development history

  time is a first-class dimension

- Intrinsically identified (SWHID, ISO/IEC 18670)

  reproducible studies: just list the SWHIDs



One global Merkle graph:
60 B nodes, 1 T edges

Point the telescope yourself

Q: Is it possible to efficiently perform software development history analyses at the scale of Software Heritage archive on a single, relatively cheap machine?

Idea

Apply graph compression techniques from the field of network analysis.

Results

The entire archive graph (~60 B nodes, 1+ T edges, June 2026) is served from ~500 GiB of RAM, traversed at the cost of tens of ns per edge (= hours for a full single-thread visit).

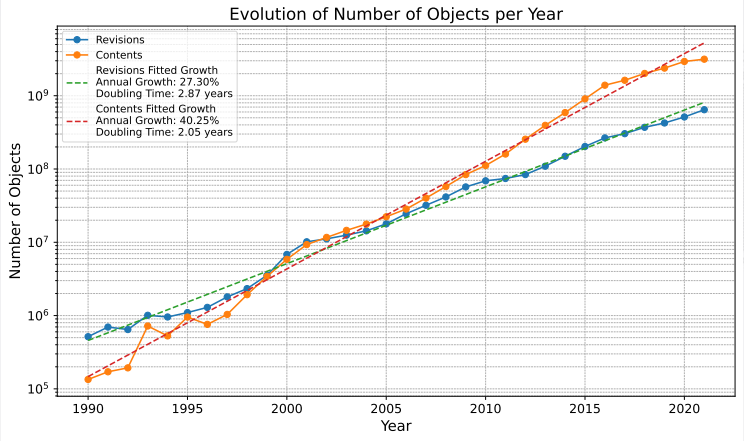


Paolo Boldi, Antoine Pietri, Sebastiano Vigna, Stefano Zacchiroli
Ultra-Large-Scale Repository Analysis via Graph Compression
SANER 2020, 27th Intl. Conf. on Software Analysis, Evolution and Reengineering. IEEE



Tommaso Fontana, Sebastiano Vigna, Stefano Zacchiroli
WebGraph: The Next Generation (Is in Rust)
WWW'24, the ACM Web Conference 2024

Rust and gRPC APIs available: docs.softwareheritage.org/devel/swh-graph/



Key findings

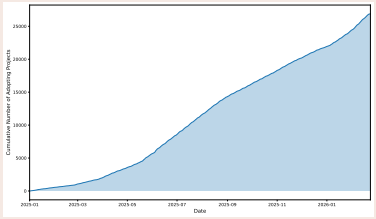
- Public code grows exponentially and has been for 30 years: original files double every ~2 years, original commits every ~3 years
- We can trace the provenance of source code at this scale in a compact relational model via the notion of isochrone graphs

Rousseau, Di Cosmo, Zacchiroli. *Software Provenance Tracking at the Scale of Public Source Code.* Empir. Softw. Eng. 25(4), 2020
Updated data & code: gitlab.softwareheritage.org/rdicosmo/archive-growth

Coding agents went from nothing to a quarter of GitHub in a year

128k projects mined for traces left by coding agents: `CLAUDE.md`, `Co-authored-by:`, `copilot/*` branches

- 22–29% of projects in Jan 2025 to Feb 2026
- Broad: every topic, every language, young and old, big and small
- AI commits are 3× larger; $\frac{2}{3}$ are features and bug fixes
- Follow-up replication study: projects *born after August 2025*: 72–76%, Claude Code leading

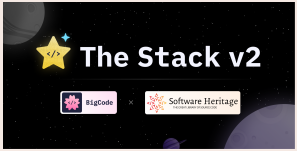


- Robbes, Matricon, Degueule, Hora, Zacchiroli. *Agentic Much? Adoption of Coding Agents on GitHub.* ACM TOSEM, 2026. Preprint: arXiv:2601.18341
- Robbes, Matricon, Degueule, Hora, Zacchiroli. *Agentic Very Much!.* Preprint: arXiv:2606.07448, 2026
- Robbes, Matricon, Degueule, Hora, Zacchiroli. *Promises, Perils, and (Timely) Heuristics for Mining Coding Agent Activity.* MSR 2026

Open, responsible, and transparent AI

only possible with: universal and identified

The Stack v2



- The Stack v2: a subset of Software Heritage, 600+ languages
- Used to train the StarCoder2 open code LLMs
- Every training file has a SWHID: *"what was the model trained on?"* has an answer

CodeCommons

CodeCommons

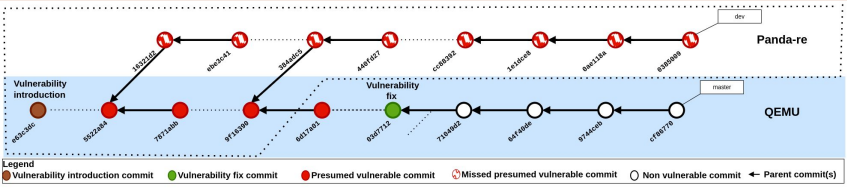
- Enrich the Software Heritage archive with project context metadata
- Leverage them to power traceable AI-generated code accompanied by rich metadata about licenses, provenance, known vulnerabilities, …

codecommons.org

Lozhkov *et al. StarCoder 2 and The Stack v2: The Next Generation.* 2024. arXiv:2402.19173

1.7 M forks left behind when a security fix lands

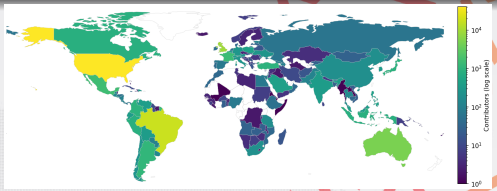
Fork between the *introduction* and the *fix* of a vulnerability, and you inherit it — silently.



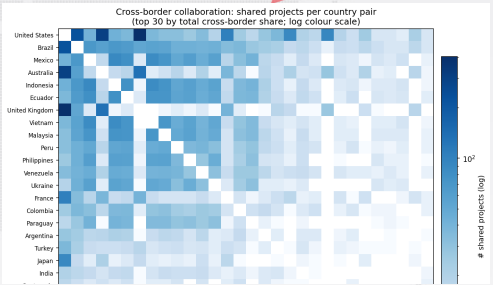
- "Color" the graph with OSV.dev data: 7162 repos → 1.7 M potentially vulnerable forks
- 86.3 M potentially vulnerable commits specific to forks, *invisible to dependency-based tooling*
- 152 cases checked by hand: 135 high-severity vulnerabilities, 9 confirmed by maintainers
- Only doable with a global, deduplicated data model: all forks, all platforms, one graph

Lefeuvre, Reux, Zacchiroli, Barais, Combemale. *Chasing One-day Vulnerabilities Across Open Source Forks.* SCORED 2026. Preprint: arXiv:2511.05097

- 193 UN member states, 288 K projects *with* government contributors, 93 K government contributors
- Public code contribution tracks e-government development (UN EGDI, $\rho = 0.52$) twice as strongly as GDP per capita ($\rho = 0.26$)
- 75% of projects have no detectable licence; 94.5% sit on a single commercial platform (GitHub); citation metadata nearly absent
- Method: filter on commit authors/ committers by email domain; swap in *your* institution, sector, or country



government contributors per country (log scale)



cross-border collaboration: shared projects per country pair

State of Public Code: How Governments Contribute to Open Source. UNESCO, UN ODET, Software Heritage, 2026. zenodo.org/records/21258432

The telescope is open

come and observe!

Software Heritage gives you

- All public code, with its full history
- In one uniform, deduplicated graph
- With intrinsic identifiers for reproducible studies
- And the heavy lifting done: datasets, compressed graph, APIs

What's next: research infrastructure

- An observatory: indicators on any slice of the software commons
- AI corpora with provenance and licences + dataset factory (CodeCommons)
- Now a French national research infrastructure (Open Science Strategy 2026-2033)

What will you point it at?

- Results: softwareheritage.org/publications
- Data: archive.softwareheritage.org / datasets.softwareheritage.org
- Contact: Stefano Zacchiroli / stefano.zacchiroli@telecom-paris.fr / @zacchiro@mastodon.xyz



Appendix

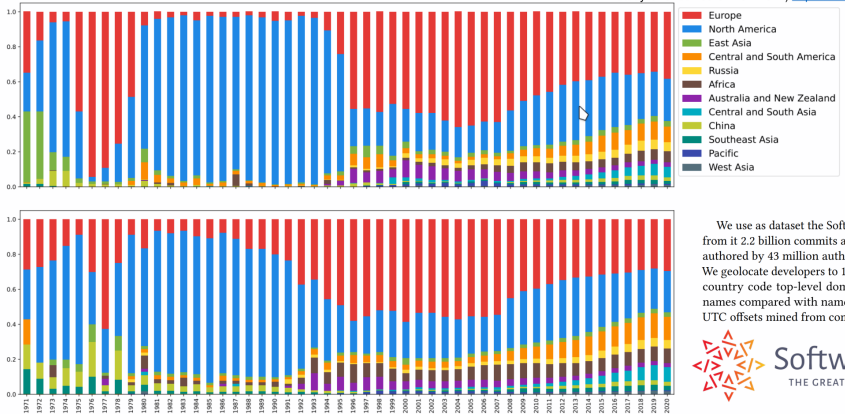
Where is public code written?

50 years, geolocated

MSR '22, May 23–24, 2022, Pittsburgh, PA, USA

Davide Rossi and Stefano Zacchiroli

Geographic Diversity in Public Code Contributions: An Exploratory Large-Scale Study Over 50 Years. MSR 2022) https://doi.org/10.1145/3524842.3528471



We use as dataset the Software Heritage archive [3] and analyze from it 2.2 billion commits archived from 160 million projects and authored by 43 million authors during the 1971–2021 time period. We geolocate developers to 12 world regions, using as signals email country code top-level domains (ccTLDs) and author (first/last) names compared with name distributions around the world, and UTC offsets mined from commit metadata.

Software Heritage
THE GREAT LIBRARY OF SOURCE CODE

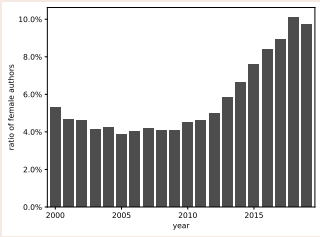
Figure 3: Ratio of commits (above) and active authors (below) by world zone over the 1971–2020 period.

Rossi, Zacchiroli. *Geographic Diversity in Public Code Contributions: An Exploratory Large-Scale Study Over 50 Years.* MSR 2022. doi:10.1145/3524842.3528471

Archived commit metadata contains public information that can be mined to study diversity, equity, inclusion (DEI) traits of the global population of public code authors.

Gender gap — key findings

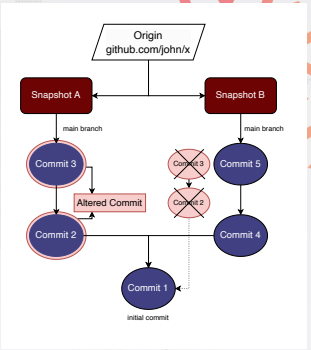
- Male authors contributed 92% of public code commits up to 2019.
- Female authors (and their commits) have grown stably for 15 years reaching 10% of yearly commits in 2019.
- The COVID-19 pandemic has *caused* a trend inversion (it is not just correlation!)



- Zacchiroli. *Gender differences in public code contributions: a 50-year perspective.* IEEE Software, 2021
- Rossi and Zacchiroli. *Worldwide gender differences in public code contributions […].* ICSE SEIS, 2022
- Casanueva et al. *The Impact of the COVID-19 Pandemic on Women's Contribution to Public Code.* Empir. Softw. Eng. 30(25), 2025

Altered histories in public Git repositories

- Git lets you rewrite published history (via force push), and forges keep no trace of it
- SWH compares successive snapshots of 111 M repos: 1.22 M repos (~1.1%) with altered histories, 8.7 M altered commits, 11.4% of them on *main/master* branches
- ~800k license files retroactively altered on main branches (32k repos)
- 13 M removals of secrets (keys, passwords): rewriting history is not security, copies persist
- Prototype tool (*GitHistorian*): vet a repository's track record of history alterations



Solal Rapaport, Laurent Pautet, Samuel Tardieu, Stefano Zacchiroli
Altered Histories in Version Control System Repositories: Evidence from the Trenches
ASE 2025 https://arxiv.org/abs/2509.09294

Software Hash Identifiers (SWHIDs)



An emerging standard

- Adoption: SPDX 2.2, IANA-registered "`swh:`" URI prefix, WikiData P6138, …
- Breaking news: standard ISO/IEC 18670:2025

Examples

- Apollo 11 AGC excerpt
- Quake III rsqrt

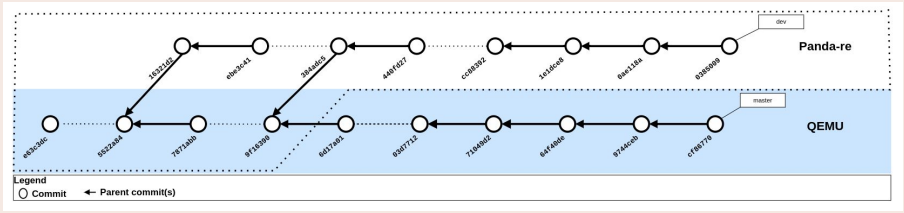
One-day vulnerabilities in open source

One-day vulnerabilities

- Def.: vulnerabilities that are publicly known, but not fixed yet in software you use.
- Challenge: identify them quickly and exhaustively, then apply countermeasures.
- Many tools available to detect one-day vulnerabilities via declared dependencies.

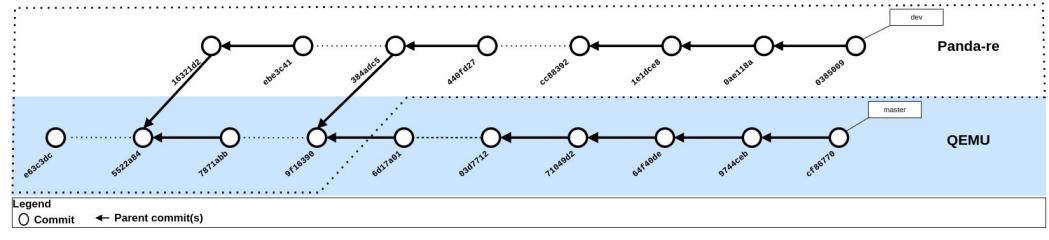
Reusing OSS via forks

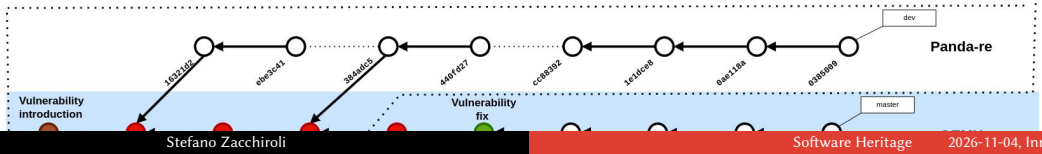
…but OSS is also reused via forking: (1) start from existing OSS (e.g., the QEMU emulator), (2) create your own (e.g., Panda-re), (3) periodically integrate changes.



Vulnerability propagation through forks

- Any change to a piece of software (*commit*) can introduce a new vulnerability.
- Or it can fix an existing vulnerability.
- What happens if a project is forked between introduction and fix of a vulnerability?
- It inherits the vulnerability, ... until the change with the fix is integrated.





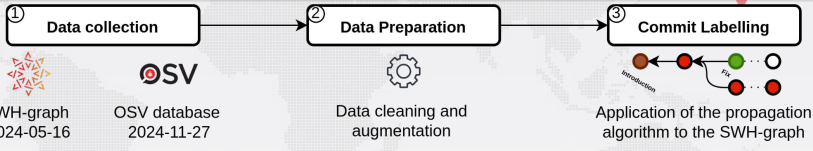
swh-vuln: chasing one-day vulnerabilities across forks… at SWH scale

Approach

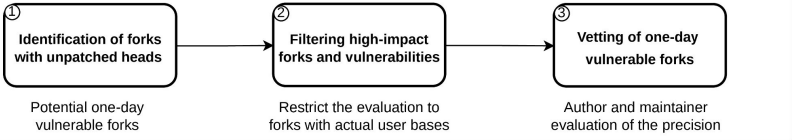
Start from a public DB of vuln. introduced/fixed in public commits (e.g., OSV.dev).

1. "Color" the entire graph of public code development history with vulnerability info.
   - Software Heritage is the only place where this can be done at the scale of all forks, across all public code, independently of specific development platforms.
2. Inform maintainers of vulnerable forks. (After validation.)

Phase 1: color the commit graph with vuln. info



Phase 2: find and vet vulnerable forks



Results (highlights)

- Starting from 7162 repos in OSV, we identified 1.7 M forks potentially vulnerable in their most recent commit.
  - 86.3 M potentially vulnerable commits specific to forks, not findable with current tools.
- We manually verified 152 cases, confirming 135 high-severity vulnerabilities in popular forks; 9 were further confirmed by maintainers.



panda-re/panda
Fork of QEMU, system emulator and virtualizer
CVE-2019-13164



go-nv/goenv
Fork of pyenv, for Go
CVE-2022-35861



sonyxperiadev/kernel
Fork of Linux, for the Xperia dev community
CVE-2021-45485
CVE-2021-4154



bitcoin-sv/bitcoin-sv
Fork of Bitcoin
CVE-2021-37492



Romain Lefeuvre, Charly Reux, Stefano Zacchiroli, Olivier Barais, Benoit Combemale
Chasing One-day Vulnerabilities Across Open Source Forks
SCORED 2026. Preprint: https://arxiv.org/abs/2511.05097