Chasing One-day Vulnerabilities Across Open Source Forks

Stefano Zacchiroli

Polytechnic Institute of Paris
stefano.zacchiroli@telecom-paris.fr

21 Oct 2025
Rencontres Cyber IP Paris
École polytechnique, Palaiseau, France

Software Heritage
THE GREAT LIBRARY OF SOURCE CODE

Outline

Open source security

Open source software can be freely used, copied, and modified.

Open Source Software (OSS) is everywhere

- Huge boost for innovation! (e.g., reduced time to market)
- 96% of (non-open) software products depend on open source (2022).
- Open source is at the heart of the global digital infrastructure.

Open source security

Open source software can be freely used, copied, and modified.

Open Source Software (OSS) is everywhere

- Huge boost for innovation! (e.g., reduced time to market)
- 96% of (non-open) software products depend on open source (2022).
- Open source is at the heart of the global digital infrastructure.

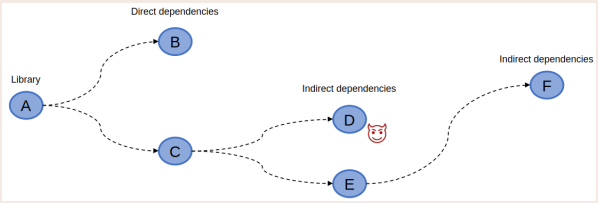
With great exposure comes great scrutiny…

- …by both good and bad actors.
- OSS is more and more targeted by attackers.
- Increased policy attention to secure OSS, e.g.:
  - US: executive orders (Biden 2022; Trump Jan 2025)
  - EU: CRA, progressively coming into effect

Software supply chain attacks

Reusing OSS via dependencies

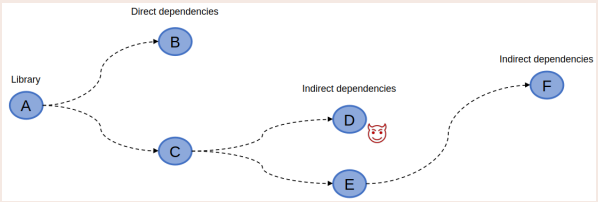
- Software dependencies: a popular way of reusing open source software.
- Software product A uses functionalities implemented in OSS product B . . . and so on.

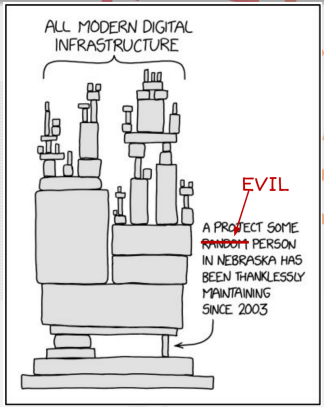


Software supply chain attacks

Reusing OSS via dependencies

- Software dependencies: a popular way of reusing open source software.
- Software product A uses functionalities implemented in OSS product B ... and so on.





based on xkcd.com/2347

Attacking the software supply chain

- Attacking undermaintained "leaf" packages (e.g., D) → efficient attack strategy
- Many documented attacks: event-stream (2018), node-ipc (2022), XZ utils (2024), ...

Outline

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

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

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

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

One infrastructure
open and shared

One infrastructure
open and shared

Cultural Heritage | Industry | Research | Public Administration

Software Heritage

The largest archive ever built

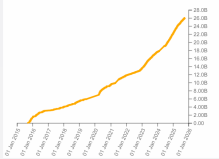
One infrastructure
open and shared

Cultural Heritage | Industry | Research | Public Administration

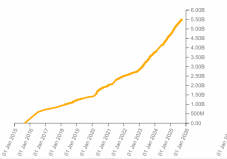
Software Heritage

The largest archive ever built

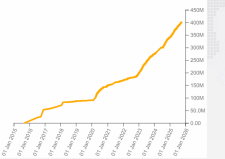
Source files
26,098,911,720



Commits
5,511,094,786



Projects
402,119,077



Bitbucket 2,818,626 origins	56,983 origins	git 33,472 origins
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	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 380,156 origins
PAGURE 72,459 origins	Phabricator 198 origins	pub.dev 63,003 origins
python 621,919 origins	SOURCEFORGE 382,368 origins	stagit 343 origins

Securing open source with Software Heritage

What does Software Heritage bring to the table?

- The largest archive that guarantees the:
  1. availability
  2. integrity → see SWHID (SoftWare Hash IDentifiers), ISO 18670 swhid.org
  3. traceability of (OSS) source code

Securing open source with Software Heritage

What does Software Heritage bring to the table?

- The largest archive that guarantees the:
  1. availability
  2. integrity → see SWHID (SoftWare Hash IDentifiers), ISO 18670 swhid.org
  3. traceability of (OSS) source code
- A universal, open knowledge base of *facts* about open source software…
- …that can be leveraged by everyone (not only the big players) to secure OSS.

Securing open source with Software Heritage

What does Software Heritage bring to the table?

- The largest archive that guarantees the:
  1. availability
  2. integrity → see SWHID (SoftWare Hash IDentifiers), ISO 18670 swhid.org
  3. traceability of (OSS) source code
- A universal, open knowledge base of *facts* about open source software…
- …that can be leveraged by everyone (not only the big players) to secure OSS.

SWHSec project

swhsec.github.io

- 2023–2027 R&D project, funded by French national CampusCyber
- 8 research teams; co-led by Télécom Paris

Axes: (1) extending SWH with security info + (2) code analysis, dependency analysis, vulnerability tracking, automatic vulnerability fixing, … at SWH scale.

Outline

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*.

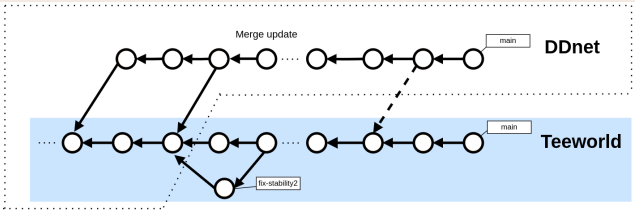
One-day vulnerabilities in open source

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*.

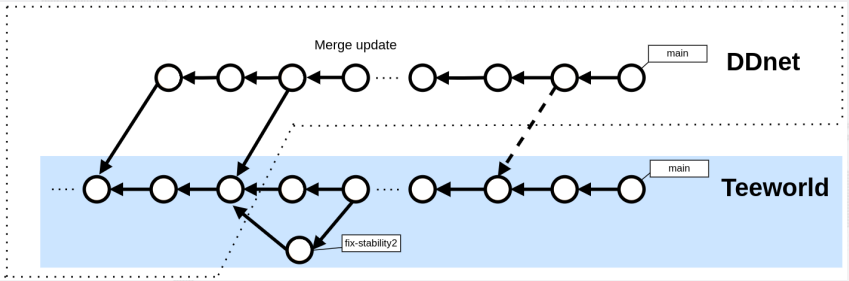
Reusing OSS via forks

Open source is also reused via forking: (1) start from existing OSS (e.g., Teeworlds game), (2) create your own (e.g., DDnet), (3) periodically integrate changes.



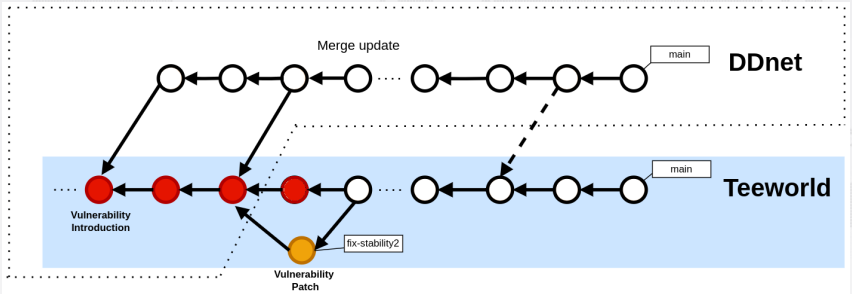
Vulnerability propagation through forks

- 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.



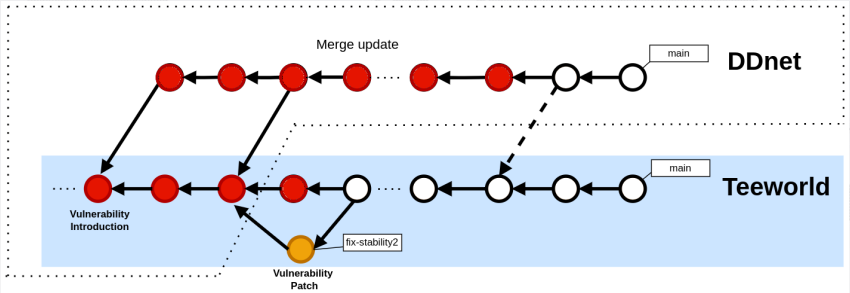
Vulnerability propagation through forks

- 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.



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

1. Start from a public DB of vuln. introduced/fixed in public commits (e.g., OSV.dev).
2. "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.
3. Inform maintainers of vulnerable forks. (After validation.)

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

Approach

1. Start from a public DB of vuln. introduced/fixed in public commits (e.g., OSV.dev).
2. "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.
3. Inform maintainers of vulnerable forks. (After validation.)

Early results

- Identified 2.2 M (million) forks of repositories referenced by OSV.dev, containing vulnerable commits; 1.3 M forks vulnerable in their most recent commit.
- 86.6 M vulnerable commits were specific to forks, not findable with current tools.
- Among 66 manually vetted cases, 5 confirmed vulnerabilities (1 critical).

Romain Lefeuvre, Charly Reux, Stefano Zacchiroli, Olivier Barais, Benoit Combemale
Chasing One-day Vulnerabilities Across Open Source Forks
To appear, 2025.

Outline

Conclusion

Software Heritage

THE GREAT LIBRARY OF SOURCE CODE

Takeaways

- Open source software is everywhere and increasingly targeted by attackers.
- State-of-the-art tooling for identifying known vulnerability is limited in scope (specific platforms, specific ways of reusing code).
- We can leverage Software Heritage to discover unfixed vulnerabilities and improve open source security for everyone. The SWHSec project is working on this.
- Next steps: integration with the Software Heritage archive, public API.

Contact

Stefano Zacchiroli / stefano.zacchiroli@telecom-paris.fr / @zacchiro@mastodon.xyz