Chasing One-day Vulnerabilities Across Open Source Forks

Romain Lefeuvre [1] Charly Reux[1] Stefano Zacchiroli[2] Olivier Barais[1] Benoit Combemale[1]

[1] *IRISA, Université de Rennes , Inria, Diverse*

[2] *LTCI, Télécom Paris, Institut Polytechnique de Paris*

Context

Open source software plays a **critical role** in digital infrastructure and fosters **code reuse**. This reuse is enabled by two main mechanisms:

1. **Dependencies**: Importing code in a separate project.

2. **Fork**: Creating a new project from an existing one.

The security of the global open-source ecosystem attracted significant attention in recent years, particularly in the wake of high-profile software supply chain attacks on specific dependencies (e.g Log4Shell).

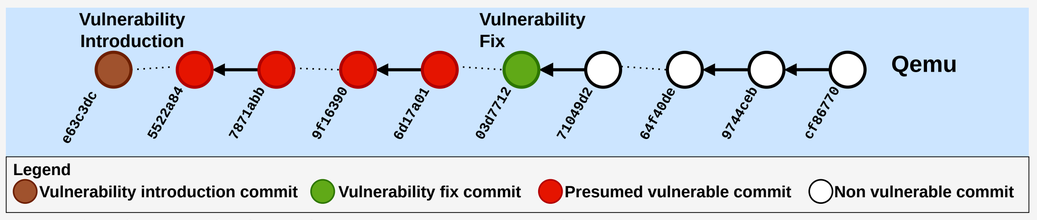
Although forking practices is widespread in open source ecosystem, it receives less attention. Forks share code history with their upstream repository but also potentially vulnerabilities, making it crucial to propagate vulnerability information to forks.

Current tooling failed in propagating vulnerability information to associated forks.

Challenge

Current approaches reason locally...

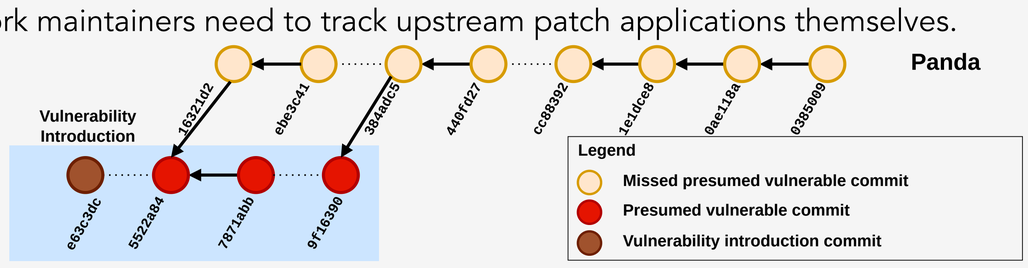
1. The impacted repository is cloned.
2. Commits between the vulnerability introduction and fix are labelled as vulnerable.



Identification of vulnerable commits associated to CVE-2019-13164 in Qemu repository

... Resulting in undetected vulnerable commits in forks

1. End users or projects depending on those fork versions are not alerted.
2. Fork maintainers need to track upstream patch applications themselves.



Missed vulnerable commits in Panda-re repository, a fork of Qemu

Forks can still remain impacted by a vulnerability for which a patch exists: A one-day vulnerability

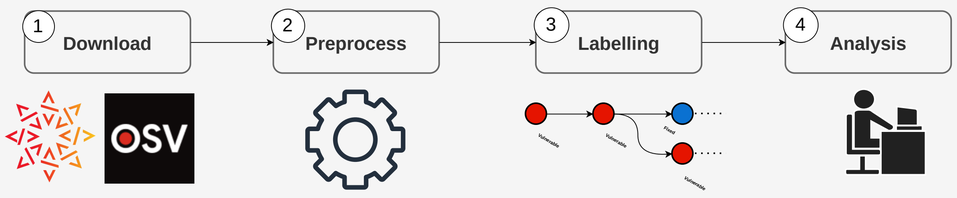
Research Methodology

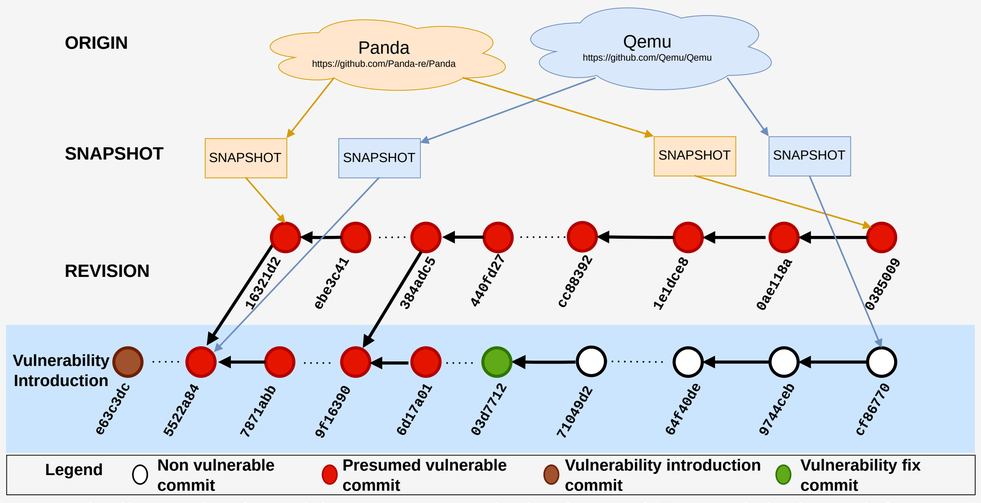
Leverage the global commit graph to track vulnerability throughout the open source ecosystem

Our approach is based on the Software Heritage (SWH) Graph, the largest public code archive, offering a global model of software source code history.

1. The SWH graph is deduplicated, a project and all its forks are linked.

2. Relying on a vulnerability database (OSV.dev) we labelled the SWH graph and propagated vulnerability information to the impacted forks.





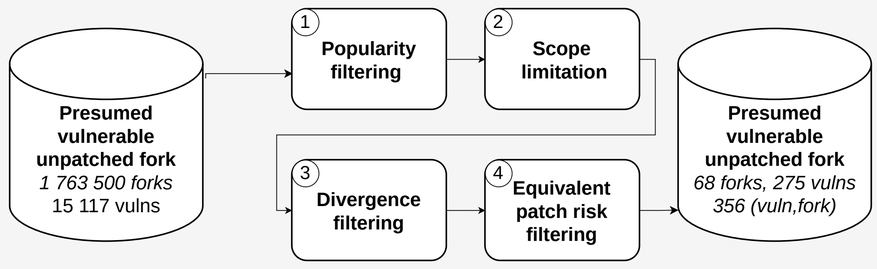
Labelling of vulnerable revision in deduplicated SWH global model

Identify one-day vulnerabilities in real-world forked open source projects

From 7162 repositories declaring a vulnerabilities in OSV.dev we detected
2.2 M of potential vulnerable forks detected

1. Most fork are not real world open source projects, for instance : git-flow forks or educational fork

2. Forks might not be impacted by upstream vulnerabilities due to divergence



Strict filtering to identify vulnerable forks

68 forks identified as potentially vulnerable **after strict filtering**

A manual evaluation has been conducted on a sample of **37 forks**

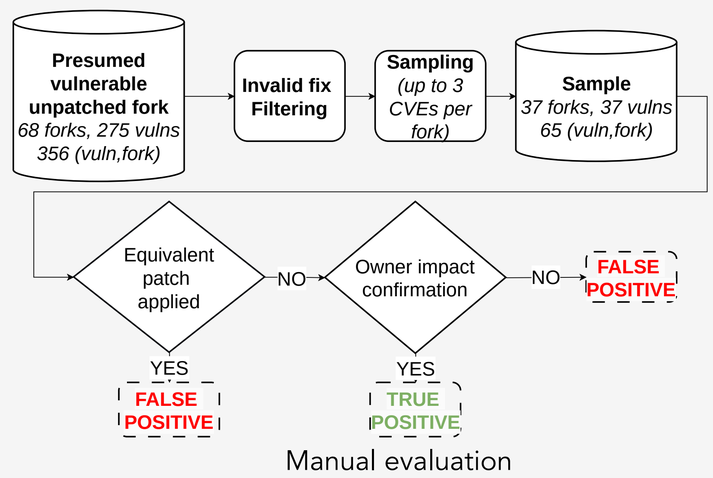
26 forks remained classified as vulnerable after manual vetting

4 one day vulnerabilities have been already confirmed as true positive by maintainer

. **panda-re/panda**, a dynamic analysis platform based on QEMU, affected by CVE-2019-13164

. **sonyxperiadev/kernel**, the Linux kernel fork by the Sony corporation, for Xperia Android devices, affected by CVE-2021-45485 and CVE-2021-4154

. **bitcoin-sv/bitcoin-sv**, a fork of bitcoin affected by CVE-2021-37492



Manual evaluation

Towards a tooling support for developers

Multi level of tooling

1. For fork project : Report on unpatched vulnerability from upstream repo

2. For project depending on a fork : Report on vulnerable fork dependencies