Google Open Source Blog

The latest news from Google on open source releases, major projects, events, and student outreach programs.

Launching OSV - Better vulnerability triage for open source

Friday, February 5, 2021

We are excited to launch OSV (Open Source Vulnerabilities), our first step towards improving vulnerability triage for developers and consumers of open source software. The goal of OSV is to provide precise data on where a vulnerability was introduced and where it got fixed, thereby helping consumers of open source software accurately identify if they are impacted and then make security fixes as quickly as possible. We have started OSV with a data set of fuzzing vulnerabilities found by the OSS-Fuzz service. OSV project evolved from our recent efforts to improve vulnerability management in open source ("Know, Prevent, Fix" framework).

Vulnerability management can be painful for both consumers and maintainers of open source software, with tedious manual work involved in many cases.

This comes from the fact that versioning schemes in existing vulnerability standards (such as Common Platform Enumeration (CPE)) do not map well with the actual open source versioning schemes, which are typically versions/tags and commit hashes. The result is missed vulnerabilities that affect downstream consumers.

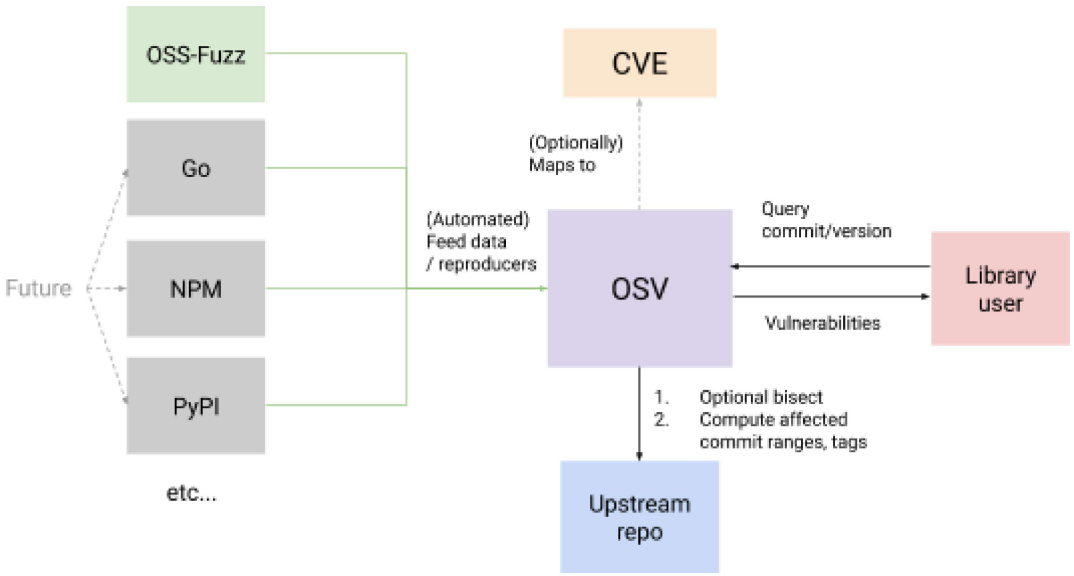
Similarly, it is time consuming for maintainers to determine an accurate list of affected versions or commits across all their branches for downstream consumers after a vulnerability is fixed, in addition to the process required for publication. Unfortunately, many open source projects, including ones that are critical to modern infrastructure, are under resourced and overworked. Maintainers don't always have the bandwidth to create and publish thorough, accurate information about their vulnerabilities even if they want to.

These challenges result in open source consumers not incorporating important security fixes promptly. OSV aims to:

1. Reduce the work required by maintainers to publish vulnerabilities, and
2. Improve the accuracy of vulnerability queries for downstream consumers by providing precise vulnerability metadata in an easy-to-query database (complementing existing vulnerability databases).

Automation

OSV aims to simplify the vulnerability reporting process for an open source package maintainer by accurately determining the list of affected versions and commits. This requires providing both the commits that introduce and fix the bugs. If that information is not available, OSV requires providing a reproduction test case and steps to generate an application build, and then it performs bisection to find these commits in an automated fashion. OSV takes care of the rest of the analysis to figure out impacted commit ranges (accounting for cherry picks) and versions/tags.



API to query for vulnerabilities. A typical OSV workflow for a package consumer looks like the picture above:

1. A package consumer sends a query to OSV with a package version or commit hash as input.

```
curl -X POST -d \
'{"commit": "6879efc2c1596d11a6a6ad296f80063b558d5e0f"}' \
'https://api.osv.dev/v1/query?key=$API_KEY'
```

```
 curl -X POST -d \
  '{"version": "1.0.0", "package": {"name": "pkg",
"ecosystem": "pypi"}' \
  'https://api.osv.dev/v1/query?key=$API_KEY'
```

2. OSV looks up the set of vulnerabilities affecting that particular version and returns a list of vulnerabilities impacting the package. The vulnerability metadata is returned in a machine-readable JSON format.
3. The package consumer uses this information to either cherry-pick security fixes (based on precise fix metadata) or update to a later version.

Ongoing work

OSV currently provides access to thousands of vulnerabilities from 380+ critical OSS projects integrated with OSS-Fuzz. We are planning to work with open source communities to extend with data from various language ecosystems (e.g. NPM, PyPI) and work out a pipeline for package maintainers to submit vulnerabilities with minimal work.

Our goal with OSV is to rethink and promote better, scalable vulnerability tracking for open source. In an ideal world, vulnerability management should be done closer to the actual open source development process, aided by automated infrastructure. Projects that depend on open source should be promptly notified and fixes uptaken quickly when a vulnerability is reported.

You can access the OSV website and documentation at https://osv.dev. You can explore the open source repo or contribute to the project on GitHub, and join the mailing list to stay up to date with OSV and share your thoughts on vulnerability tracking.

By Oliver Chang and Kim Lewandowski, Google Security Team